REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL (*if applicable*) 32	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, *State, and ZIP Code*) Monterey, CA 93943-5000	7b. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000
8a. NAME OF FUNDING / SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (*if applicable*)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, *State, and ZIP Code*)	10.SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO / PROJECT NO / TASK NO / WORK UNIT ACCESSION NO

11. TITLE *(Include security Classification)*
Performance Evaluation Of UHF Fading Satellite Channel By Simulation For Different Modulation Schemes.

12. PERSONAL AUTHOR(S) Betülhan Kahraman (TUN)

13a. TYPE OF REPORT Master's Thesis	13b. TIME COVERED FROM _____ TO _____	14. DATE OF REPORT (*Year, Month, Day*) 1992 December 10	15. PAGE COUNT 81

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect official policy or position of the Department of Defense or the U.S. Government

17. COSATI CODES			18. SUBJECT TERMS (*Continue on reverse if necessary and identify by block number*)
FIELD	GROUP	SUB-GROUP	UHF Fading Satellite Channel Simulation, UHF Follow-on Satellite, DBPSK, DQPSK, MSK, Simulated Link Margin, Convolutional Encoding.

19. ABSTRACT (*Continue on reverse if necessary and identify by block number*)

In this report, we discuss the physics of fading and summarize the methods to mitigate its devastating effects for communications channels . We evaluated performance curves of Differentially Encoded Binary Phase Shift Keying (DBPSK), Rate 1/2 Convolutionally Encoded Differential Quadrature Phase Shift Keying (DQPSK) and Minimum Phase Shift Keying (MSK), by simulation for both AWGN and fading channels. Further, we have simulated typical transponder components of a UHF satellite system and evaluated their effect for the above modulation structures. Finally, we have calculated the carrier-to-noise ratios by using simulation curves to assess link margins for the UHF Follow-on Satellite.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT [X] UNCLASSIFIED/UNLIMITED [ ] SAME AS REPORT [ ] DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBILE INDIVIDUAL Paul H. Moose	22b. TELEPHONE (*Include Area Code*) (408) 659 1938 22c. OFFICE SYMBOL EC / Me

DD Form 1473, JUN 86

Previous editions are obsolete
S/N 0102-LF-014-6603

Approved for public release; distribution is unlimited.

Performance Evaluation of UHF Fading Satellite Channel by Simulation for Different Modulation Schemes

by

Betulhan Kahraman
Lieutenant Junior Grade, Turkish Navy
B.S.E.E, Naval Postgraduate School, 1990

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

ABSTRACT

In this report, we discuss the physics of fading and summarize the methods to mitigate its devastating effects for communication channels. We evaluated performance curves of Differentially Encoded Binary Phase Shift Keying (DBPSK), Rate 1/2 Convolutionally Encoded Differential Quadrature Phase Shift Keying (DQPSK) and Minimum Shift Keying (MSK), by simulation for both AWGN and fading channels. Further, we simulated typical transponder components of a UHF satellite system and evaluated their effect for the above modulation structures. Finally, we calculated the carrier-to-noise ratios by using simulation curves to assess link margins for the UHF Follow-On Satellite.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

During the period when the HF, or so-called short-wave radio was the corner stone of long distance communications, signal fading, also termed signal scintillations, was accepted as an inevitable problem that one should live with.

This problem was thought to be solved when the *Early Bird* (INTELSAT I) was launched in April 1965 by International Telecommunications Satellite Organization (INTELSAT) [Ref. 1: p. 1].

Satellites, with their broad coverage and line-of-sight communications capabilities, enabled the communications community to utilize higher carrier frequencies. Ionospheric disturbances were thought to be negligible at these higher frequencies and were modeled using only additive white Gaussian noise (AWGN). But deployment of these earth-orbiting devices has generated reports of unexpected disturbances in the communication channels [Ref. 2: p. 5]. Because of those reports, some measurements were made all over the world, revealing that the cause for these disturbances was, in fact, ionospheric irregularities. This knowledge brought a new understanding and awareness for the importance of fading for UHF satellite communication systems. Today the importance of

fading is well appreciated by the members of the communications community.

A. FADING AND ITS EFFECTS

Before getting into the effects of fading for satellite channels and the methods to mitigate these effects, we will try to summarize the physics and causative components of fading.

Signal fading, or what radio astronomers call scintillation, is caused by signal propagation through a region which has a medium with random fluctuating index of refraction [Ref. 2: pp. 6]. These fluctuations result from atmospheric turbulance, meteorological events, dust along tropospheric paths and the ionization structure along transionospheric paths. These disturbances have two main effects over an electromagnetic signal, one of which is a random phase disturbance due to variations in the phase velocity and the other is a random magnitude fluctuation due to scattering.

Recovering the transmitted information from this disturbed signal is the main challenge for communications engineers who are in search of a fast and reliable communication system. If a communications system is not designed to operate in the presence of signal fading, the performance of the system may be severely degraded if fading is encountered. Some satellite communications systems were originally designed for operation

in the AWGN channel based on the assumption that the ionosphere would have negligible effect on carriers above 100 Mhz or so, but later UHF high frequency carrier systems were observed to be significantly affected by signal fading.

Fading can be observed in two different forms related to the symbol period. Slow fading, in which the symbol amplitude and phase variations can be observed to slowly change compared to the symbol period, has a highly random fluctuating error pattern with long bursts of errors scattered among prolonged error free periods. Fast fading, with its rapid fluctuations during the symbol period, gives slight chance for successful coherent demodulation and the bit error rate approaches 0.5 [Ref. 2: p. 7].

Since the motion of the ions and index of refraction changes are random to the channel user, these random variations can be treated in statistical terms. Fading statistics also can be encountered in two different forms according to the physical environment of the propagation medium. A Rayleigh fading channel model is the most common multipath propagation model. In this model, signal fading is caused by time-variant multipath characteristics of the channel, thus forming a channel impulse response which can be modeled as a zero mean complex-valued Gaussian process with a Rayleigh distributed envelope [Ref. 2: p. 7]. When the medium has fixed scatterers or signal reflectors in addition to the randomly moving scatterers, the channel can no longer be

modeled as having a zero mean. The resultant channel is the Ricean distributed fading channel.

Also important is frequency selectivity. If the coherence bandwidth of the channel is very small in comparison to the bandwidth of the transmitted signal, the channel is said to be frequency selective [Ref.3: pp. 707-708]. Otherwise, it is non-frequency selective. When fading is observed in the frequency selective form, the distortion of the signal is very severe compared to the nonselective form.

B. FADING MITIGATION METHODS

There are two major methods for mitigating the damaging and undesirable effects of fading on communications systems.

1. Modulation Scheme Selection

The first solution is directly related to the modulation/demodulation method. With selection of a modulation scheme such as differentially coherent phase shift keying (DPSK), which does not require carrier phase tracking, instead of conventional coherent phase shift keying (CPSK), which is an optimum modulation scheme for AWGN channel, communications systems are less suspectable to the rapid phase disturbances. When a modulation scheme is selected in which the channel bit rate does not exceed the minimum value of the frequency selective bandwidth, the system is said to be working in its slow fading limit [Ref. 2: p. 17]. This arrangement in the modulation/demodulation scheme improves performance to a limit

which still may not be acceptable with normal link power margins of the system, but is better than the fast fading performance of CPSK. In this event, a second mitigation step must be invoked.

2. Use of Error Correction Coding and Interleaver

This second step incorporates error correction coding and diversity techniques to improve average error rates in a slow fading environment.

a. Error Correcting Coding

The basics of error correcting coding were first introduced by Shannon in 1948. Since then, a great deal of effort has been expended on the problem of devising efficient encoding and decoding methods for error control due to channel noise.

The objective of coding is to utilize redundant data to retrieve information during the decoding phase without increasing the average power. This is done by increasing the effective minimum distance between the signals. One way to do this is to use convolutional codes. Convolutional codes have proved useful in mitigating effects of fading particularly when combined with interleaving or other forms of diversity techniques. When provided with a reasonable error pattern, these codes, in conjunction with soft decision Viterbi decoding, are effective in reducing binary error rates from 10^{-2} to 10^{-5} or less. In Chapter II we provide the detailed

specification of the rate 1/2 convolutional encoder we implemented.

b. Diversity Techniques

Diversity techniques use the concept of redistributing extensively long error bursts to present a more random-appearing error pattern to the decoder. These techniques split modulator output in a controlled manner and recombine after the channel to break the effect of burst fading patterns into shorter groups of errors for the decoder to accommodate [Ref. 2: p. 208]. Diversity techniques are effective for the slow fading environment in which the fading pattern can be observed in long burst forms.

The block interleaving method uses a two-dimensional memory array. Data is entered into the matrix row by row and fetched out columnwise. The inverse operation (deinterleaving) takes place in the receiver before the demodulation process. The details of the block interleaver implemented for this project are presented in Chapter II.

Also important in mitigating the effects of fading for the satellite channel is finding the optimum modulation scheme in regard to bandwidth and power limitations. The typical satellite channel has the constraints of both bandwidth and power. One way to increase the power efficiency is to use error correcting coding by adding extra bits to the transmitted symbol sequence. This approach will decrease the

bandwidth efficiency by forcing the modulator to operate at higher data rates which results in occupation of a larger bandwidth. For increasing the bandwidth efficiency, a higher order modulation scheme can be selected to maintain the same error probability; this will result in larger signal power consumption which is not power efficient [Ref. 4: p. 67]. Therefore, an optimum scheme must be found.

Recent research on this problem emphasized the concept of combined coding-modulation schemes. In his important paper [Ref. 5], Ungerboeck states the advantages of Trellis coded modulation (TCM) as a tool that achieves significant coding gains over conventional uncoded multilevel modulation without compromising bandwidth efficiency. To implement this, one uses a convolutional encoder to increase power efficiency and finds a higher order signaling scheme to retain the bandwidth efficiency. For example, instead of using uncoded 4-PSK we could use a rate 2/3 convolutional encoder cascaded with an 8-PSK modulation scheme that will give the same throughput with some coding gain but without expanding the bandwidth required for 4-PSK.

If we take into account the non-linear high power amplifiers which are generally used for satellite channels, we are led to consider chosing constant envelope modulation schemes. Continuous phase modulation has two attributes that make it attractive for the satellite channel: constant envelope and low spectral occupancy [Ref. 4:p. 240]. The

application of coding to this modulation scheme adds memory to its own virtual memory to improve energy efficiency. This combined convolutional coding and continuous phase modulation structure is of particular interest for energy constrained and band limited nonlinear satellite channels.

Coded continuous phase modulation (CPM) was the starting point for this project. Existing satellite channels of interest use BDPSK modulation. We were to design a system that would have the same performance with a higher throughput or better performance with the same throughput. We chose a QPSK modulation scheme integrated with a conventional rate 1/2 convolutional encoder. Both systems were tested through the fading satellite channel at different data rates. In addition we implemented minimum shift keying (MSK) modulation, a special form of CPFSK with modulation index h=1/2, through the satellite channel to compare its performance with those of the above two schemes.

We simulated the UHF Satellite channel by using the program PRO_MATLAB on a SUN SPARK Station. The channel was under the effect of flat fading and AWGN. We used typical space segment components like a prefilter, hardlimiter and postfilter to simulate the satellite transponder.

Chapter II provides detailed specifications of the UHF satellite channel and its space segment components. Chapter III presents the simulation results for DBPSK, rate 1/2 convolutional encoded QPSK and minimum shift keying (MSK) for

different data rates. Detailed link budget calculations and conclusions are given in Chapters IV and V, respectively. Simulation codes are included as appendices .

II. UHF SATELLITE CHANNEL AND SPACE SEGMENTS

A. FLAT FADING MODEL

In this thesis we have used the Rayleigh fading channel model which is outlined in detail in Ref. 1, and shown in Fig. 1.

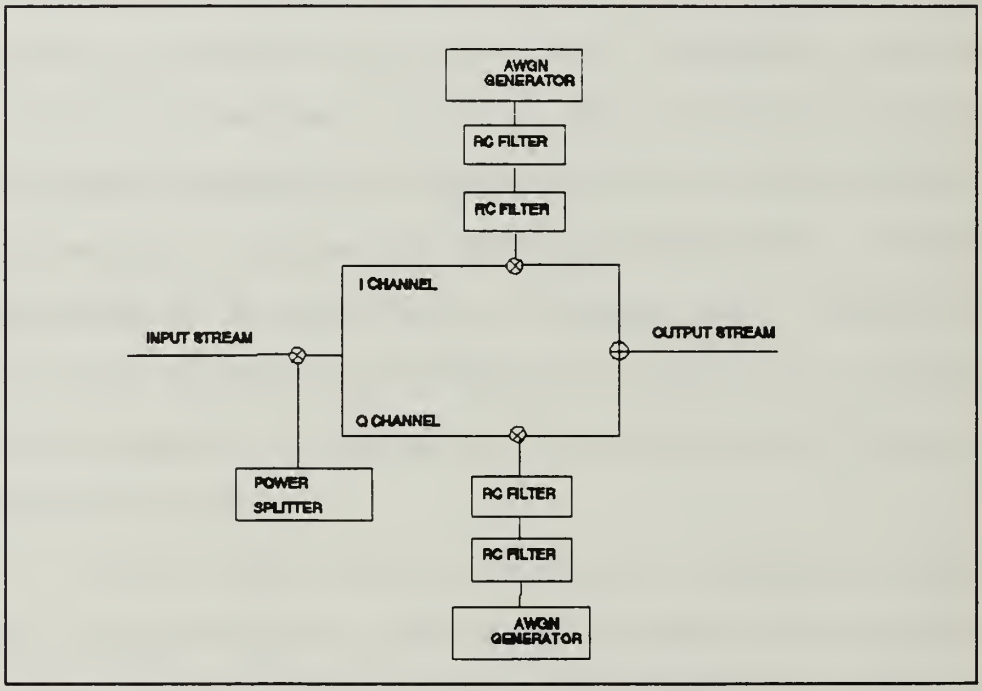

Figure 1 Rayleigh Flat Fading Model

The basis of the Rayleigh flat fading model, which is actually an implementation of a nonselective fading channel, is to impose a complex modulation in both amplitude and phase on the transmitted signal $S_T=m(t)e^{j\omega ct}$. Then we can represent our real received signal as

$$S_R(t) = Re(A(t)\, e^{j\theta(t)} m(t)\, e^{j\omega_c t}) \qquad (2\text{-}1)$$

where A(t) and θ(t) are time-varying amplitude and phase generated by the channel, m(t) is the transmitted data modulation and ω_c is the carrier angular frequency. This representation assumes that the fading spectrum is narrowband with respect to the carrier frequency [Ref.2: p. 52]. Furthermore, (2-1) can be represented in two quadrature modulating waveforms as follows

$$S_R(t) = I(t)\,[m(t)\cos\omega_c t] + Q(t)\,[m(t)\cos(\omega_c t + \frac{\pi}{2})] \qquad (2\text{-}2)$$

where

$$I(t) = A(t)\cos\theta(t) \qquad (4)$$

$$Q(t) = A(t)\sin\theta(t)$$

These equations show that we can produce a flat fading received signal by passing the transmitted signal through a power splitter in case of non-complex working conditions, then shifting the phase of one of the outputs by 90 degrees and passing this output through two product modulators. The quadrature modulating waveforms $I(t)$ and $Q(t)$ are the channel disturbances. To simulate the channel accordingly, we have to define their statistics.

As indicated in Ref. 1, for the Rayleigh case these I and

Q modulating waveforms should be statistically independent and each should be zero-mean Gaussian process with equal variances. For our simulation purposes we have used this model. For the DPSK case at baseband, the transmitted signal was real. We multiplied this transmitted signal with $e^{j\pi/4}$ to have I and Q channels with equal power. For the other modulation schemes we worked in complex form so no power splitter was needed. After we had our I and Q channels split from the transmitted signal, for each channel we produced white Gaussian noise independent of each other and passed these Gaussian noise sequences through two cascaded RC filters whose specifications are defined in Ref. 2 as follows.

Each RC filter had a time constant $\tau_F=RC=\tau_0/2.146193$. These filters have been implemented by the simple difference equation

$$y_k=ay_{k-1}+bx_k \tag{2-3}$$

where y_k is the output and x_k is the input of the filter. The filter coefficients a and b were determined by filter time constant and the filter gain respectively. When the filter power gain is set to unity, the values of a and b become

$$a=e^{\left(\frac{-\Delta t}{\tau_F}\right)} \tag{2-4}$$

$$b = \left[\frac{(1-a^2)^3}{(1+a^2)}\right]^{1/4} \qquad (2\text{-}5)$$

where Δ_t is the sampling time interval. These values of a and b affirm the resultant fading signal to have unit variance and zero mean. As stated in Ref. 1, the value for Δ_t should be less than or equal to $0.1\tau_0$ which is concluded experimentally. Therefore filter coefficient *a* has values between 0.2146193 and 0.02146193. In our project we used 0.2146193 for fast fading and 0.02146193 for slow fading.

The resulting fading signals are multiplied with the *I* and *Q* channel of the transmitted signal separately to produce the fading effected received signal.

Figures 3 and 4 show the effect of slow and fast fading for a random generated input stream shown in Fig. 2.

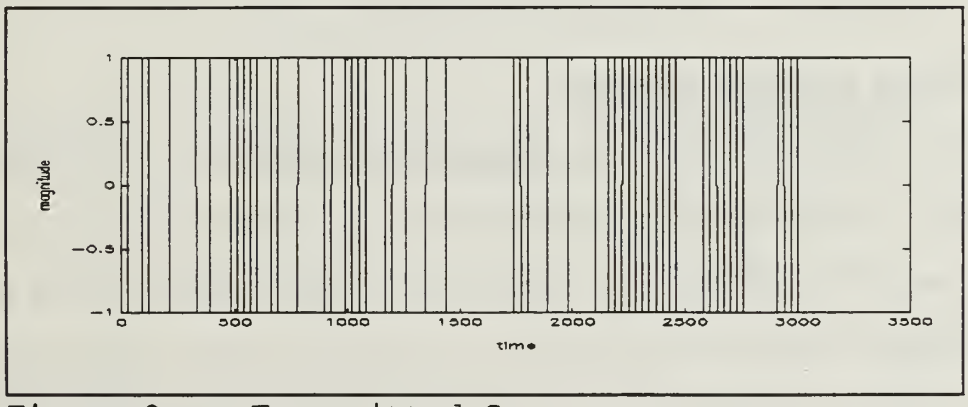

Figure 2 Transmitted Sequence

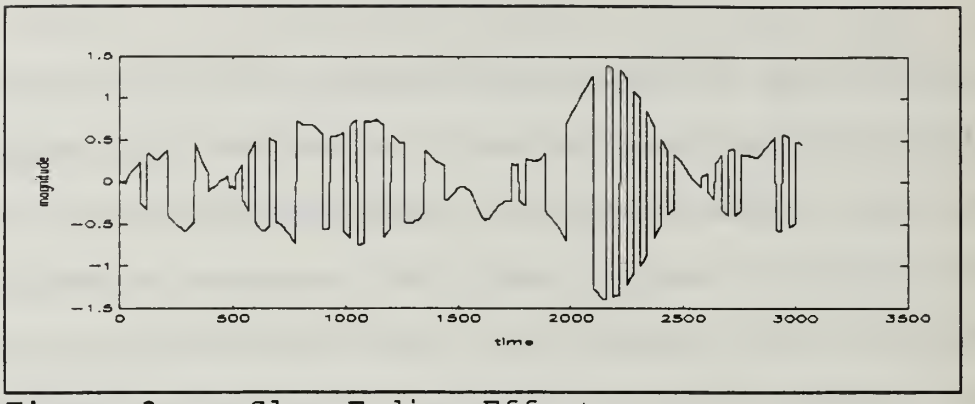

Figure 3 Slow Fading Effect

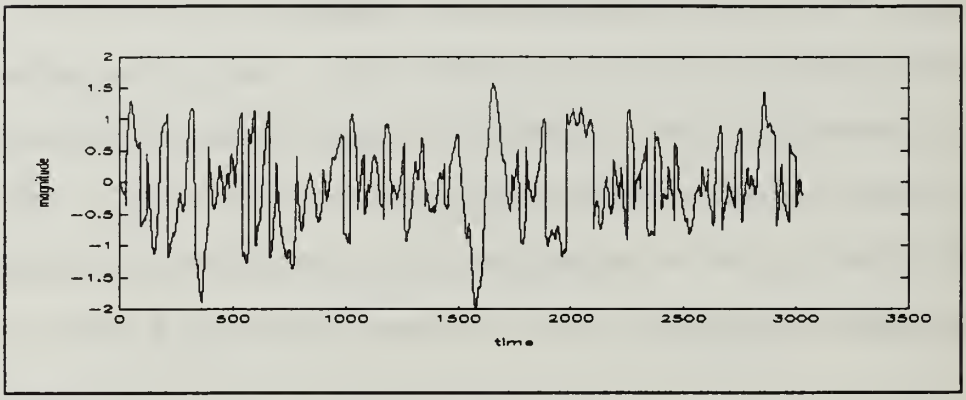

Figure 4 Fast Fading Effect

B. SPACE SEGMENT MODULES

In addition to the fading simulation, we simulated typical space segment modules for a satellite channel. The modules used during the simulation were adopted from Ref. 6. The modules were a prelimiter filter, a hard limiter and a postlimiter filter for a typical 25 kHz channel.

We will explain the main functions and detailed specifications of each below.

1. Prelimiter Filter

The purpose of this filter is to establish the desired channel bandwidth and provide an approximate 60 dB rejection for the stopband of 15 kHz from the corner frequency. The filter is implemented as Chebyshev Filter Type I with 6 poles and for 0.01 dB passband ripple. The magnitude and phase plots of the frequency response and step response of this filter are shown in Figs. 5, 6 and 7, respectively.

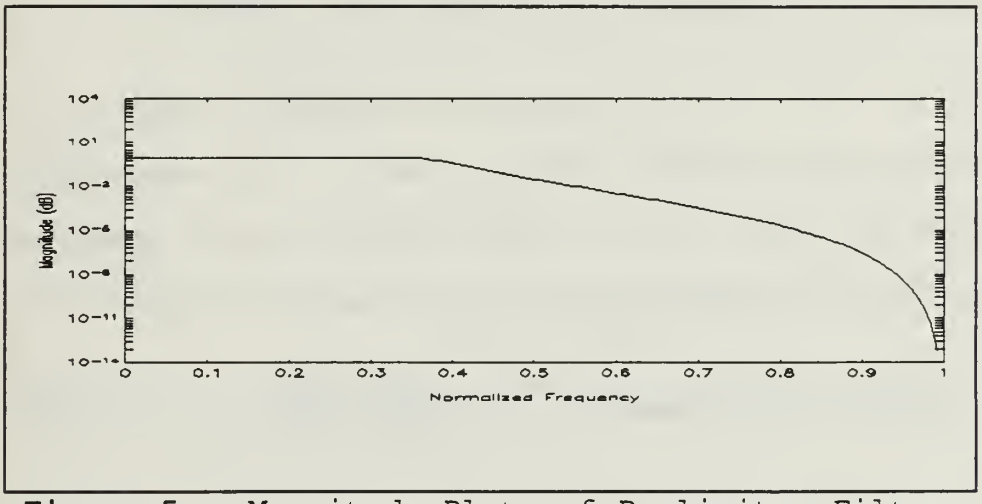

Figure 5 Magnitude Plot of Prelimiter Filter Frequency Response

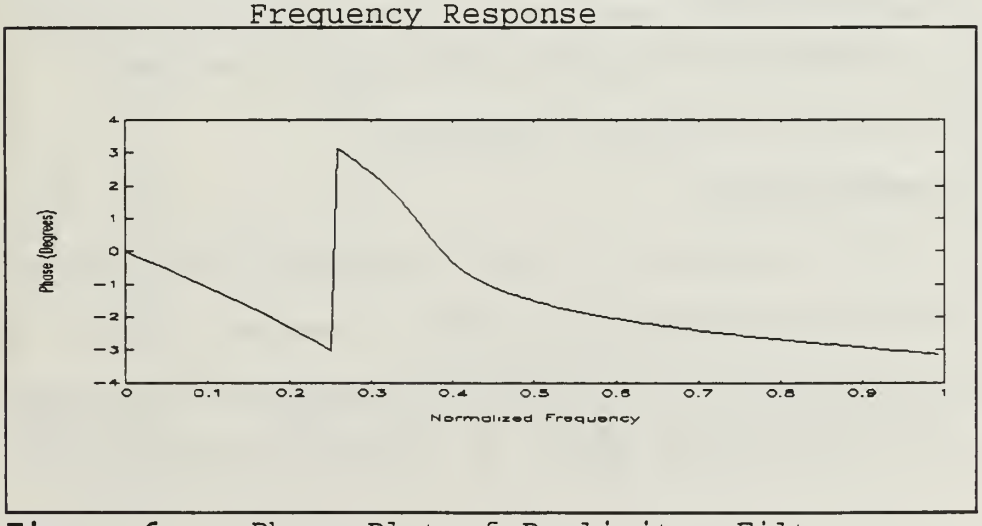

Figure 6 Phase Plot of Prelimiter Filter Frequency Response

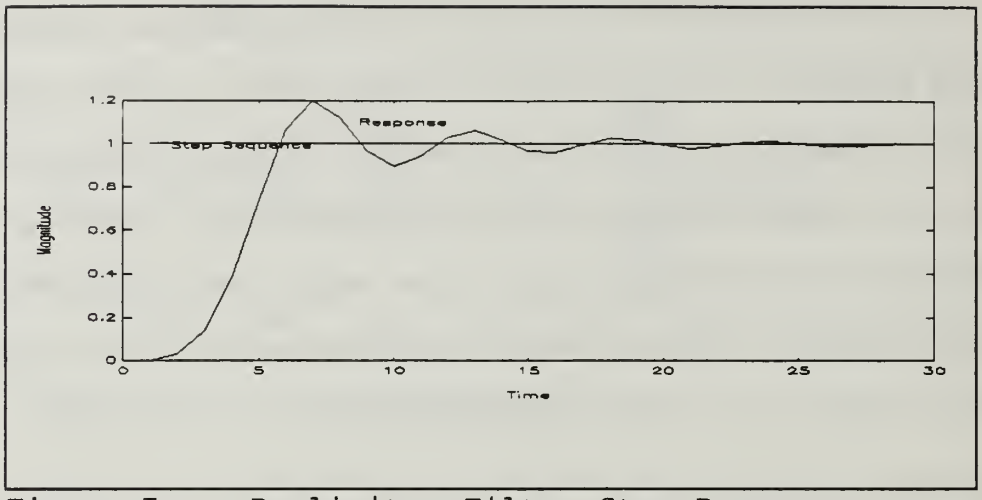

Figure 7 Prelimiter Filter Step Response

Filter function of the simulation program accepts the cutoff frequency value between 0 and 1, where 1 corresponds to half the sampling rate. For 12.5 kHz analog cutoff frequency the digital cutoff frequency was calculated to be 0.34 from

$$f_{CUTOFF(normalized)} = \frac{2f_c}{r_b N} = \frac{2f_c}{f_s} \qquad (2\text{-}6)$$

where r_b : Baud rate,

N : Number of discrete time samples per baud'

f_s : Analog sampling frequency

2. Hardlimiter

The hardlimiter is used to provide constant output power from the threshold of noise to maximum signal input. This was simulated by dividing each sample by its magnitude such that each complex sample is on the unit circle.

3. Postlimiter Filter

The postlimiter filter attenuates out-of-band limiter noise power and spectral regrowth due to limiter and rejects multipath induced frequency response ripple. It was implemented as a Chebyshev Type I filter with 4 poles and 0.025 db passband ripple. The magnitude and phase plots of the frequency response and the plot of the unit step response for this filter are shown in Figs. 8, 9 and 10, respectively.

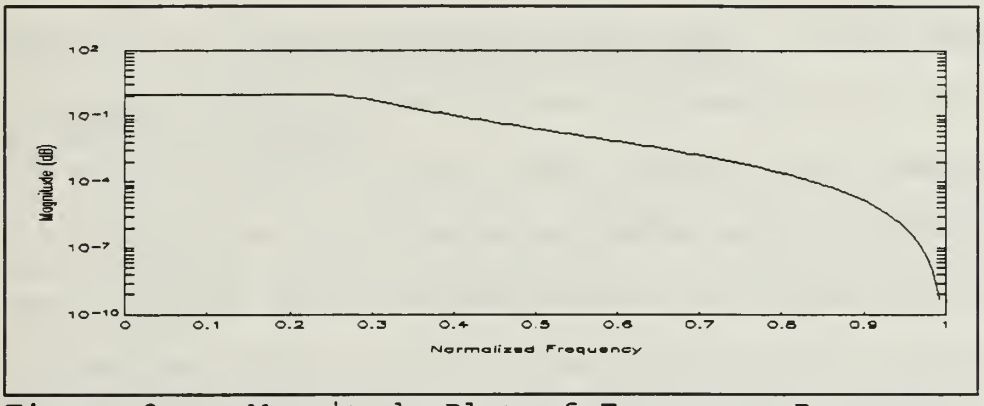

Figure 8 Magnitude Plot of Frequency Response of The Postlimiter Filter

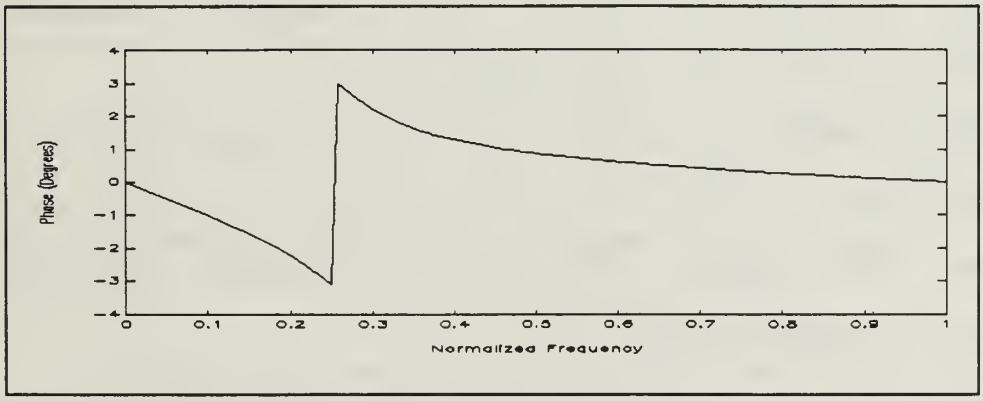

Figure 9 Phase Plot of Frequency Response of The Postlimiter Filter

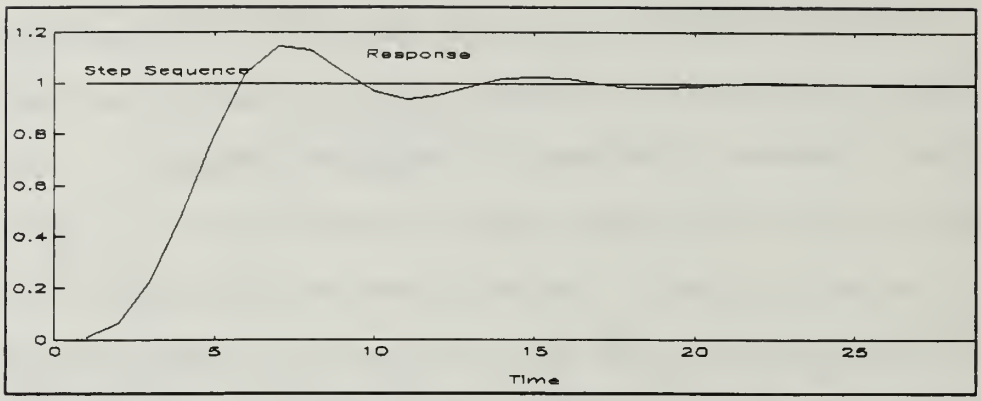

Figure 10 Step Response of Postlimiter Filter

Amplitude and phase responses of the channel are primarily determined by these filters. The low passband ripple requirement has imposed tight constraints over these filters; as a result, phase delays have occurred. An equalizer filter could be build to counterbalance this effect, but with simulation tools we applied a certain amount of delay to the sample stream to have the same approximate effect. Because of the hard limiter, the channel amplitude response is dominated by the postlimiter filter. Phase response, however, is dominated by both pre- and post-filters.

With summaries of the flat fading model and the space segment units we concluded the presentation of the satellite channel. The overall UHF Satellite channel implementation diagram is shown in Fig. 11.

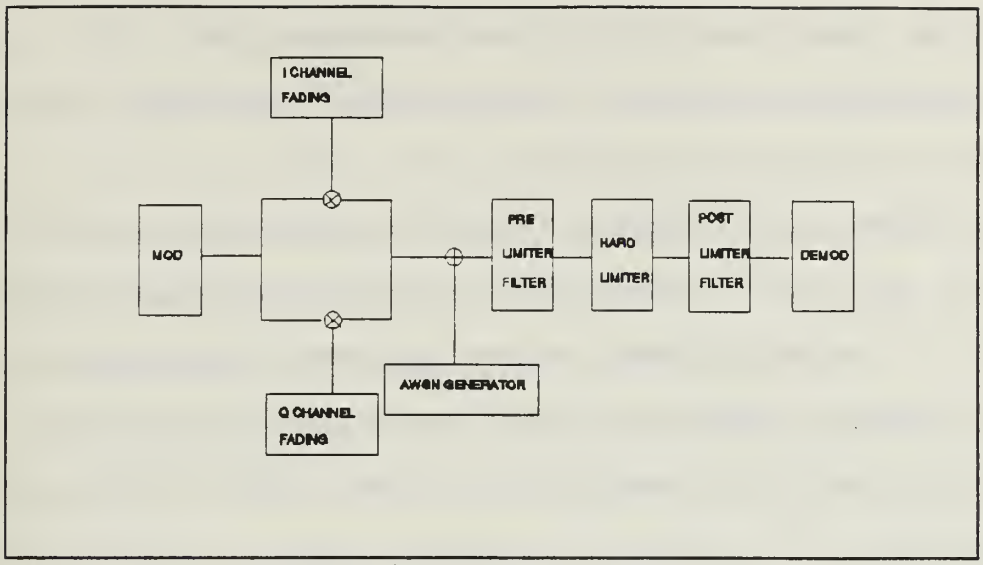

Figure 11 UHF Fading Satellite Channel

C. MODULATION SCHEMES

In this section the basic concepts of the modulation schemes that were tested with the UHF satellite fading channel are summarized.

1. Differential Binary Phase Shift Keying (DBPSK)

One of the important considerations in the modulation design is hardware simplicity. DBPSK modulation discipline does not require phase coherency and thus allows a relatively simple receiver structure due to absence of a synchronization circuit [Ref. 7: p. 546]. This virtue makes it popular for channels where the phase shift changes slowly compared to the bit duration.

Differential encoding is achieved by adding a reference bit at the beginning of the data stream. If the bit which comes after the reference is the same as the reference

bit, no phase difference takes place. If the bit has a different value compared to the reference bit, 180 degrees phase difference applies; and so forth for each bit with respect to the preceding bit.

There are primarily three decoding methods for the DBPSK as described in Ref. 8. We have simulated the bit-by-bit method of decoding DBPSK which theoretically requires, at most, 1 dB more Eb/No than that for Quadrature-PSK (QPSK) or BPSK provided that probability of error is 10^{-4} or less [Ref. 7: pp. 546].

Our implementation of DBPSK decoding was to take the difference and the sum of the current and previous symbol and compare the absolute values to determine the decision metric. If this metric is less than 0, then decision is made in favor of 0; otherwise 1.

The modulator and demodulator implementation for this signaling scheme is shown in Fig. 12.

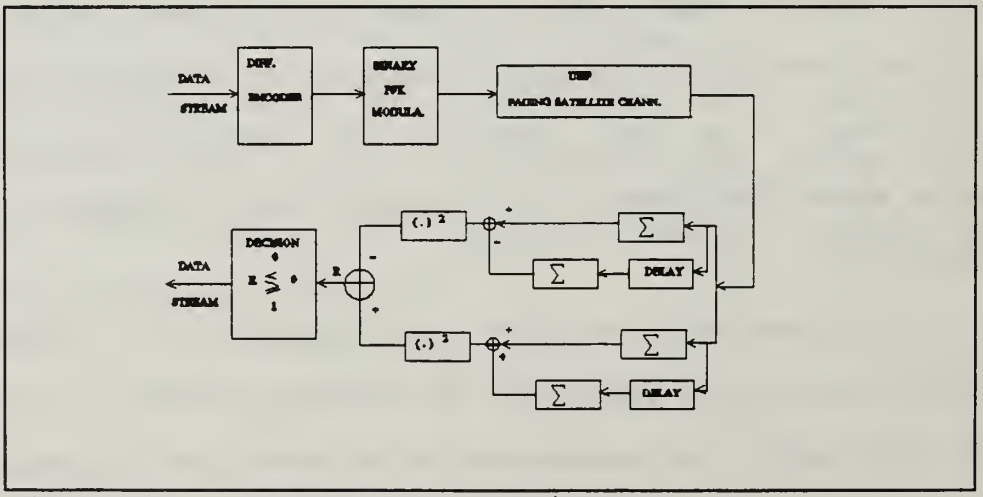

Figure 12 DBPSK Modulator/Demodulator

We have taken this modulation technique as our base modulation scheme for the satellite channel simulation, since it is currently widely employed. Codes related with DBPSK are enclosed in Appendix A. We will present their performance in AWGN and fading channels in Chapter III.

2. Rate 1/2 Convolutionally Encoded QPSK.

We selected convolutionally encoded quadrature phase shift keying (QPSK) modulation scheme as an alternative for BDPSK, since it has the same throughput and bandwidth as BDPSK. The convolutional encoder adds one more code bit for each bit in the data stream but, level 4 mapping of QPSK balances this excess bit so that the system has the same throughput and bandwith as DBPSK. Most important is that due to its convolutional encoded structure, this combined modulation scheme will be more immune than DBPSK to the destructive effects of the fading when blended with an interleaver.

a. Rate 1/2 Convolutional Encoder

Convolutional codes are named after the linear convolution that takes place between the information sequence and the code generator matrix to form the codewords [Ref. 9: p. 1].

The encoder is linear and a finite state machine that has a sequential process ability. Therefore, it is

generally implemented by using shift registers. The number of memory cells in the shift registers plus one is equal to its constraint length K . The rate of the convolutional encoder is defined as the ratio of the number of the input bits divided by the corresponding output bits.

In this project we have used the standard rate 1/2 convolutional encoder which has $(133,171)_8$ connections. The convolutional encoder and its connections are illustrated in Fig. 13.

The maximum likely sequence estimator decoder for these codes uses the Viterbi Algorithm [Ref. 10]. The algorithm works iteratively symbol by symbol, tracing through the trellis to find the maximum likelihood correct path [Ref. 8: p. 147]. The algorithm uses Euclidian distance branch metrics to compare each of the paths through the trellis for N symbol intervals (search depth) in the received vector.

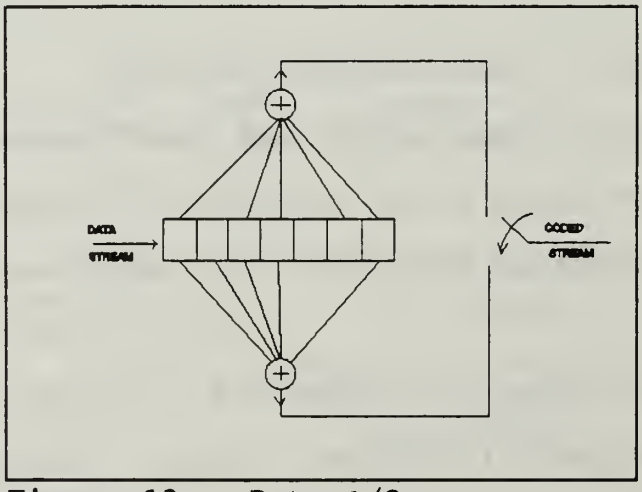

Figure 13 Rate 1/2 Convolutional Encoder

b. Differential Quadrature Phase Shift Keying (DQPSK)

QPSK modulation is a modulation scheme that uses 4 phases per symbol. Thus, two bits are transmitted during each signaling interval.

QPSK signal is equivalent to two orthogonal BPSK signals and it occupies the same bandwidth as that of a BPSK signal.

For a satellite channel we desire a modulation scheme that does not require a synchronization circuit, nor a noncohorent modulation scheme. For this reason we have encoded our QPSK signal differentially, as explained in DBPSK case. The performance of our implementation of DQPSK was 3 dB below the optimum matched filter coherent QPSK receiver [Ref. 7: p. 548].

This combined convolutional encoded differential QPSK modulation scheme finally is blended with a block interleaver/deinterleaver to mitigate the effects of the fading channel by spreading out the error bursts in time .

The codes related to QPSK modulation simulation are in Appendix B.

c. Block Interleaver

The block interleaver/deinterleaver for this application was basically a two-dimensional matrix with *k* rows *and l* columuns where *k.l=Total number of symbols*. The number of rows (*k*) determines the degree of the interleaver and is

set long enough for slow fading to ensure the errors within a code word are independent. However, the delay caused by this remapping in real modem implementations must be taken into consideration.

The block diagram of the overall implementation for the combined structure is shown in Fig. 14.

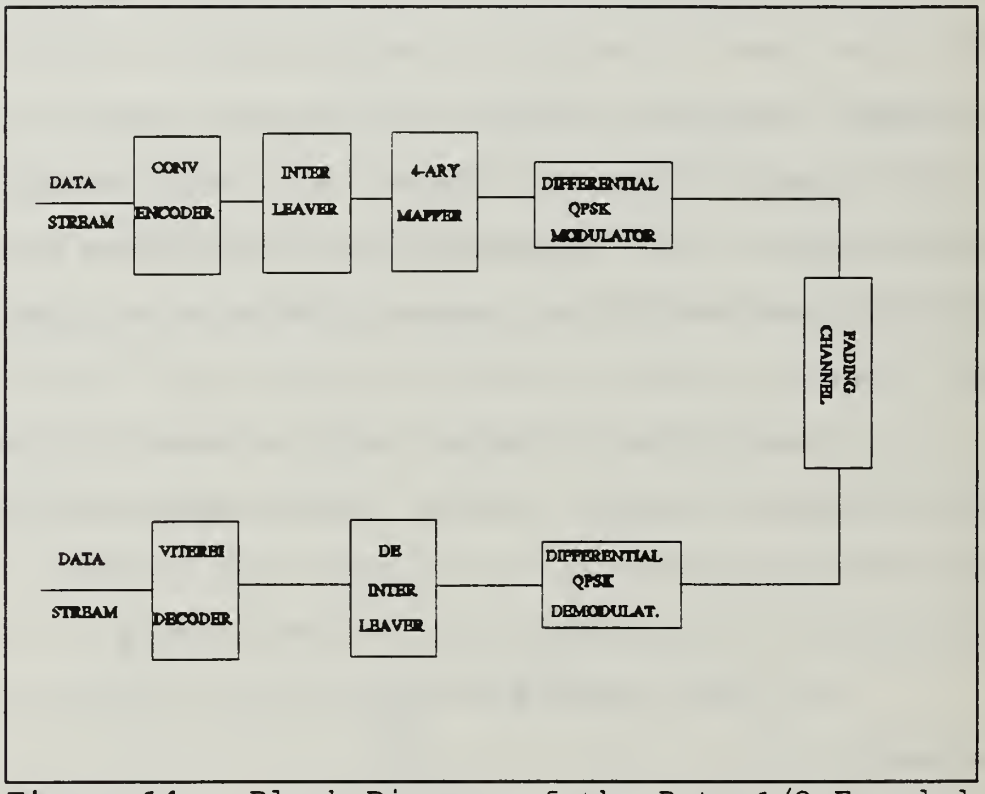

Figure 14 Block Diagram of the Rate 1/2 Encoded DQPSK Modulation Structure

3. Minimum Shift Keying (MSK)

As we stated in Chapter I, there is an interest in constant envelope modulation schemes in the satellite channels due to nonlinear high power amplifiers. Continuous-phase

shift keying (CPFSK) modulation has a constant envelope structure and restricted spectral width. We simulated this modulation scheme and passed it through fading channel to have a brief understanding of its spectral efficiency in the satellite channel and its response under fading conditions.

The continuous-phase shift keying signal is generated by feeding the data signal into a frequency modulator. The output of the frequency modulator will have signal shape described by [Ref. 3: p. 173].

$$s(t) = A\cos[2\pi f_c t + \Phi(t;I) + \phi_0] \qquad (2\text{-}7)$$

where

A : Signal amplitude,

f_c : Carrier frequency,

$\phi(t;I)$: Time-varying phase of the signal,

ϕ_0 : Initial phase,

t : Time,

I : Modulation value.

The time-varying phase of the signal can be described as

$$\phi(t;I) = 2\pi f_d T \sum_{k=-\infty}^{n-1} I_k + 2\pi f_d q(t-nT) I_n ; nT < t \leq (n+1)T \qquad (2\text{-}8)$$

or by equating $h(\text{modulation index}) = 2f_dT$ we can rewrite the above equation as

$$\phi(t;I) = \pi h \sum_{k=-\infty}^{n-1} I_k + 2\pi f_d I_n q(t-nT) \qquad (2\text{-}9)$$

where

f_d : Peak frequency deviation,

T : Symbol duration,

n : Symbol number,

I : Modulation value,

q(t) : Phase shaping waveform.

The phase shaping waveform is the integral of the frequency shift, that is,

$$q(t) = \int_0^t u(\tau)\, d\tau \qquad (2\text{-}10)$$

The frequency pulse $u(t)$ has a finite duration for the interval $0 \le t \le LT$. L is the pulse length and determines the response of the modulation. L=1 is called full response CPFSK, and L>1 is partial response CPFSK. Although frequency pulses have discontinuities, the integration makes the phase-shaping pulse continuous, and the resulting waveform has a continuous phase which is a function of both time and modulation values. In (2-8) $I = \ldots I_{-2}, I_{-1}, I_0, I_1, \ldots$ denotes the modulation sequence sent to the CPFSK modulator. I_n takes values $\{\pm 1, \pm 3, \ldots, \pm(M-1)\}$ where $M = 2^k$ *for M-ary CPFSK*, and k corresponds to number of information bits per symbol. The parameter h *(modulation index)* determines the number of states in CPFSK. If h is a

rational number, then the number of states in CPFSK is finite. It is common practice to select a modulation index

$$h=\frac{p}{q} \qquad (2\text{-}11)$$

where p and q are relatively prime integers. These integers determine the phase state so that when p is even, there are q sates and 2q states when p is odd [Ref.3: pp. 172-175].

CPFSK signals are generally characterized by their phase trajectory versus time. These phase trajectories can be represented in a trellis structure, which for the binary sequence is shown in Fig. 15.

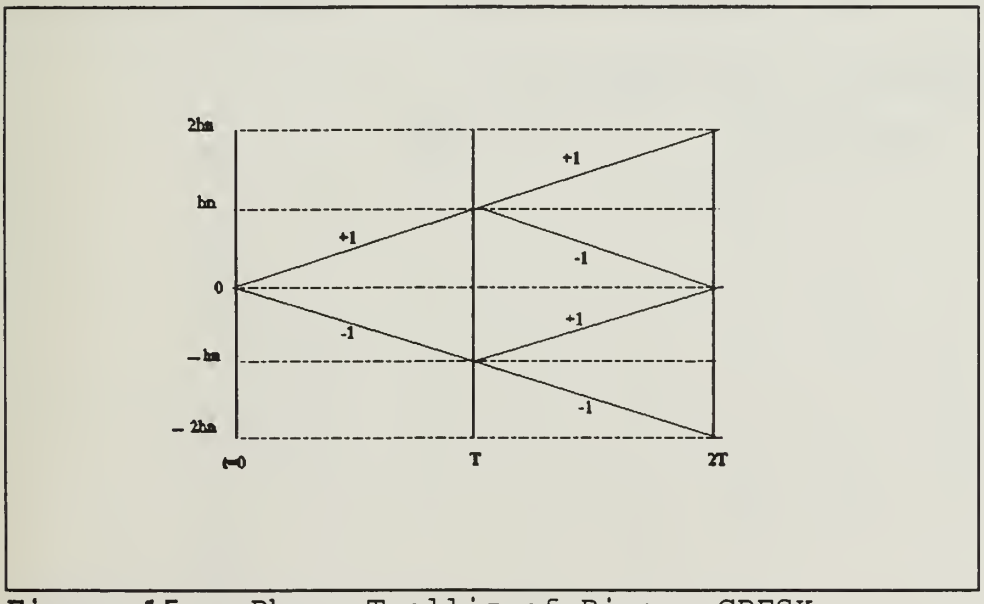

Figure 15 Phase Trellis of Binary CPFSK

Changing three parameters of the CPFSK structure, modulation index, pulse shape and the alphabet size of the bit stream, enables one to generate variety of CPFSK signals. One important of these is MSK signaling, which is binary CPFSK modulation with index $h=1/2$. MSK modulation uses minimum frequency separation of $\Delta f=1/2T$ to ensure that the signaling waveforms are orthogonal [Ref. 3: pp. 180].

In our simulation, the MSK waveform was produced by implementing (2-6) for baseband signal generation with zero carrier frequency, as described in [Ref. 7: pp. 351-353]. The demodulator used for the simulation was an implementation of the FSK demodulator for an MSK waveform [Ref. 8: pp. 258-259]. In the demodulator we used two match filters to extract the binary signals and applied their output to the Viterbi algorithm for soft decoding. Appendix C contains simulation programs related to MSK modulation.

III. SIMULATION RESULTS

In this chapter we present the simulation results for DBPSK, DQPSK and MSK modulations in the fading satellite channel.

A. DBPSK SIMULATION RESULTS

The simulation for DBPSK modulation was conducted in three categories.

1. AWGN Channel Simulation

We used 10000 bits to simulate the base DBPSK modulation and compared this with the theoretical curve to make sure our simulation setup was accurate. The results are illustrated in Fig. 16.

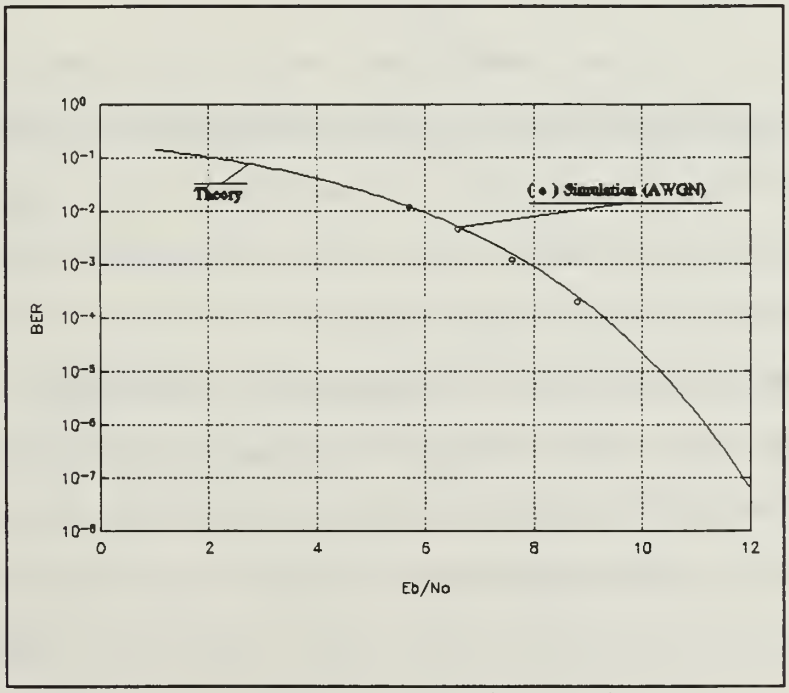

Figure 16 Theory and Simulation Comparison for AWGN Channel

The simulation results agree with the theoretical results. The theoretical curve was plotted using [Ref. 7: pp. 549]

$$P_e = \frac{1}{2} e^{-(\frac{E_b}{N_o})} \qquad (3\text{-}1)$$

We plotted our simulation points as the ratio of number of decoding errors out of the total number of transmitted data bits versus E_b/N_o , where

$$(\frac{E_b}{N_o}) = \frac{\sum_{n=1}^{n=N} (A)^2}{2^k . \sigma^2} \qquad (3\text{-}2)$$

and A is the signal amplitude, N is the number of samples during a symbol period, k is the number of bits per symbol and σ is the standard deviation of the AWGN. In our implementation the values chosen were $A \in \{1, -1\}$, $N=30, k=1$. The σ varied according to the E_b/N_o under test.

2. AWGN Channel and Transponder Components

In this test, we have included the prelimiter filter, hard limiter and postlimiter filter portions of the satellite transponder to determine their effect on the performance. We conducted the simulation for 2400 baud with the satellite

transponder components included. The results are shown in Fig. 17.

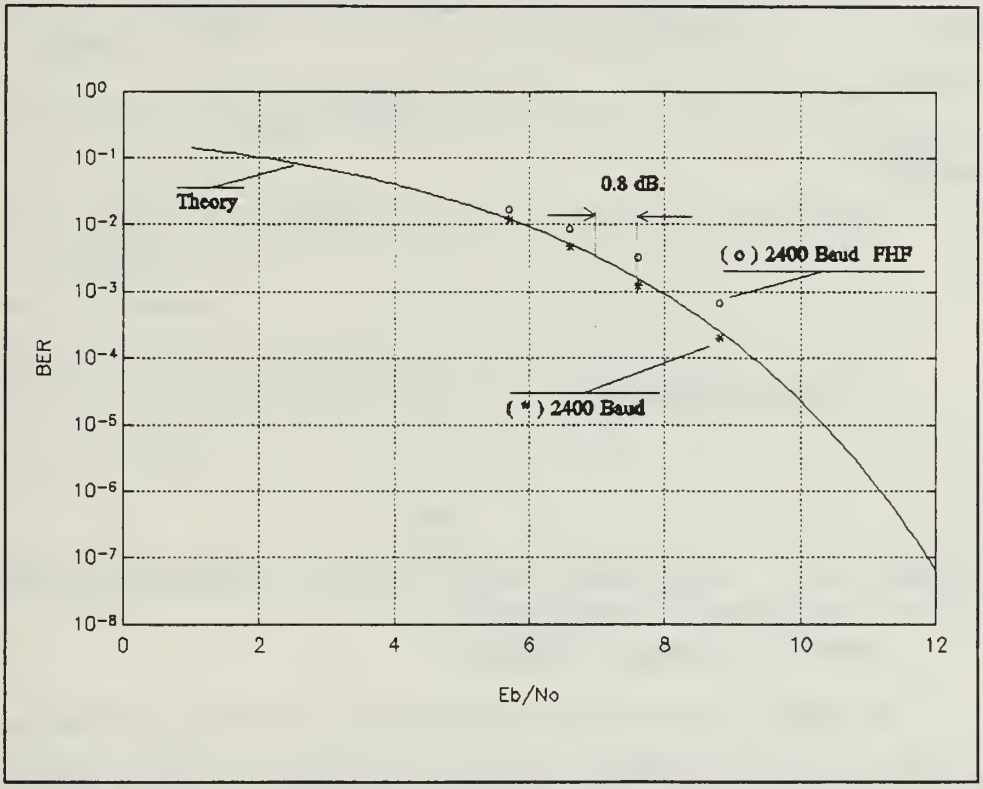

Figure 17 Effect of Transponder Components

The data show that the effect of inclusion of transponder components in the channel is approximately *0.8 dB* loss in performance. We used the same setup to see the response of 4800, 9600 and 19200 baud data rates by changing the number of samples per symbol and corresponding noise standard deviation. These data are shown in Fig. 18. There is almost constant 1 dB loss for each data rate increase.

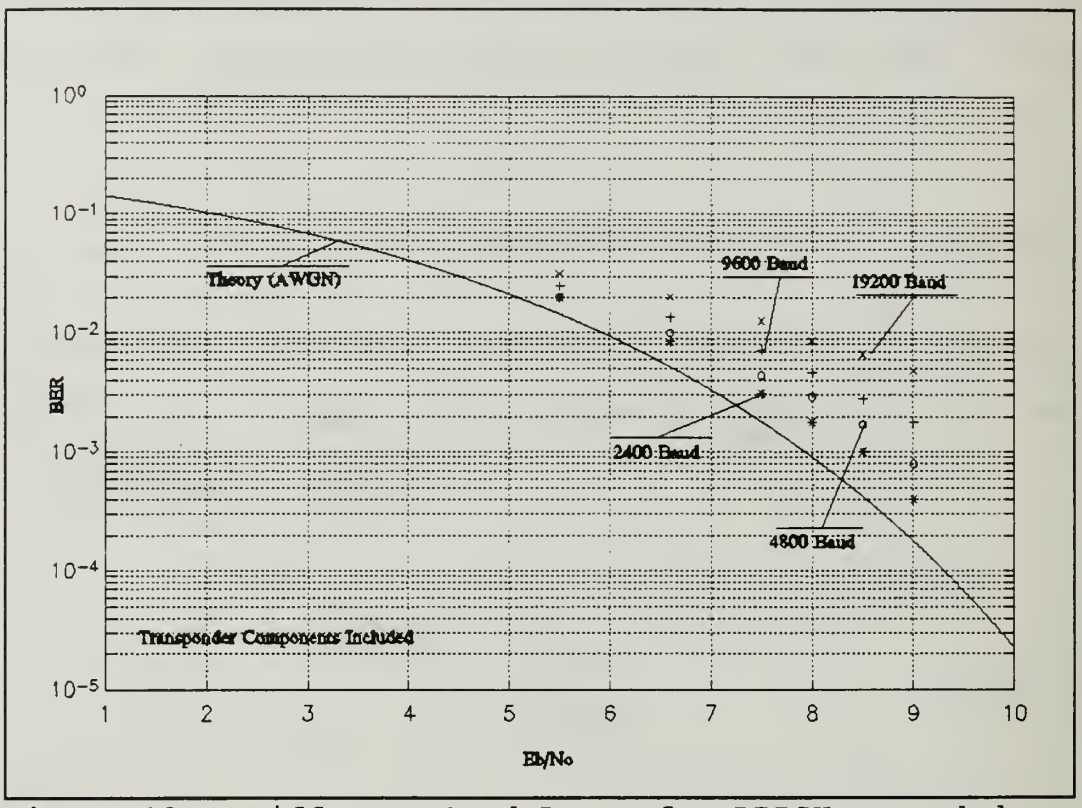

Figure 18 Different Baud Rates for DBPSK cascaded with Transponder Components

3. Satelite Fading Channel

We used the fading generator in the slow fading mode as explained in Chapter II for this simulation. We tested four transmission rates through this channel as shown in Fig. 19.

If we compare the theoretical performance for AWGN channel with the 2400 baud fading effected performance we see a catastrophic degradation which requires higher signal power for a desired bit error rate. This is an unacceptable solution for satellite channels, due to physical constraints of the space segment.

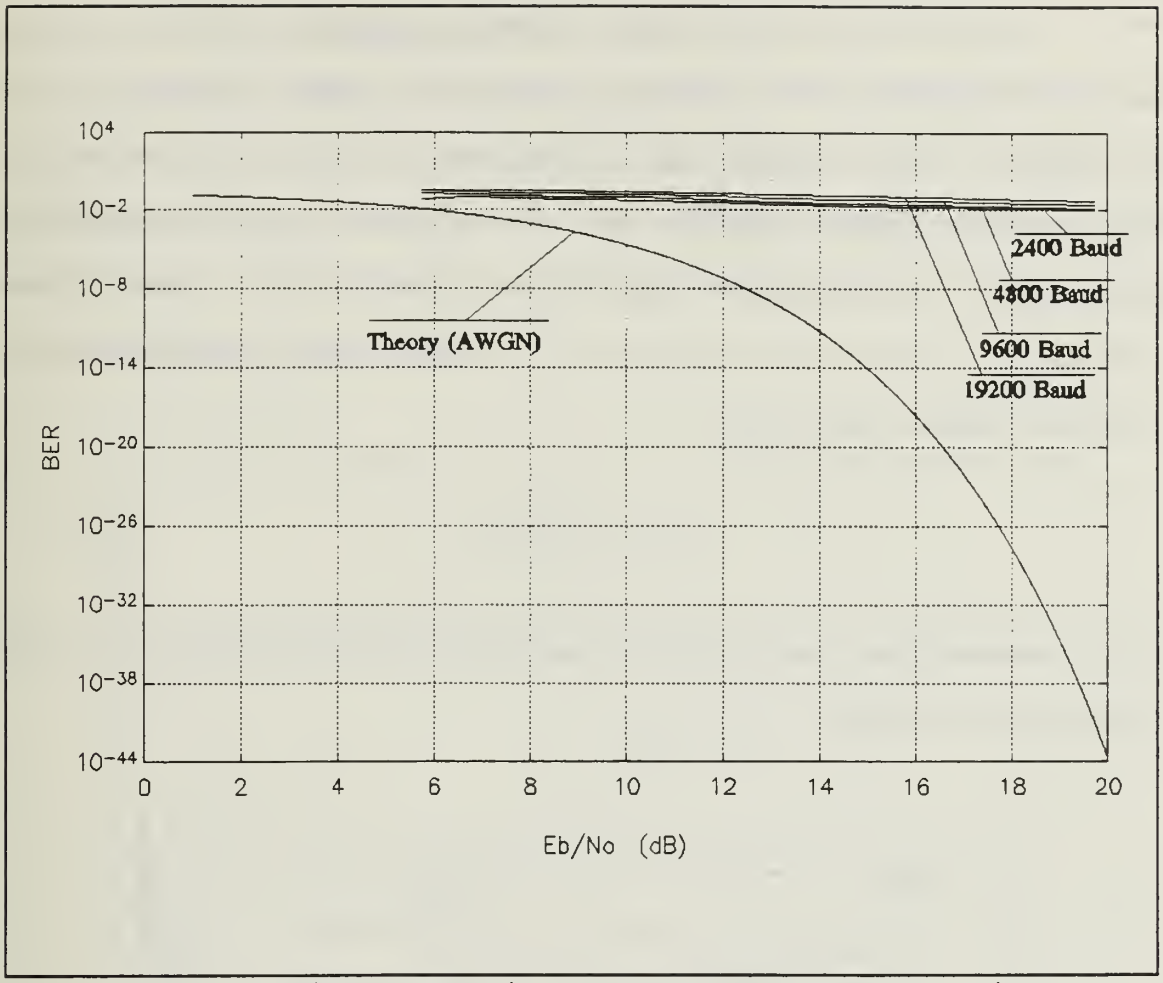

Figure 19 Fading Satellite Channel Effect over Different Baud rates for DBPSK Modulation.

B. DQPSK SIMULATIONS

Considering the simulation results for the fading satellite channel and noting that the energy requirement is very high for desired bit error rate, we proceeded to implement rate 1/2 convolutional encoded DQPSK modulation for the channel. Simulations related to DQPSK were performed in the following categories.

1. DQPSK Simulation in AWGN Channel

To be sure that our DQPSK simulation codes were working properly, we checked DQPSK in AWGN channel. The outcome of this simulation is plotted in Fig. 20. For this simulation we have used k=2 in (3-2) for the relation between E_b/N_o and the standard deviation (σ) of the AWGN. The theoretical curve was plotted 3 dB above the performance of coherent QPSK, which is

$$P_e = Q\left(\sqrt{2\left(\frac{E_b}{N_o}\right)}\right) \qquad (3\text{-}3)$$

The figure shows that simulation results matched well with the theoretical curve.

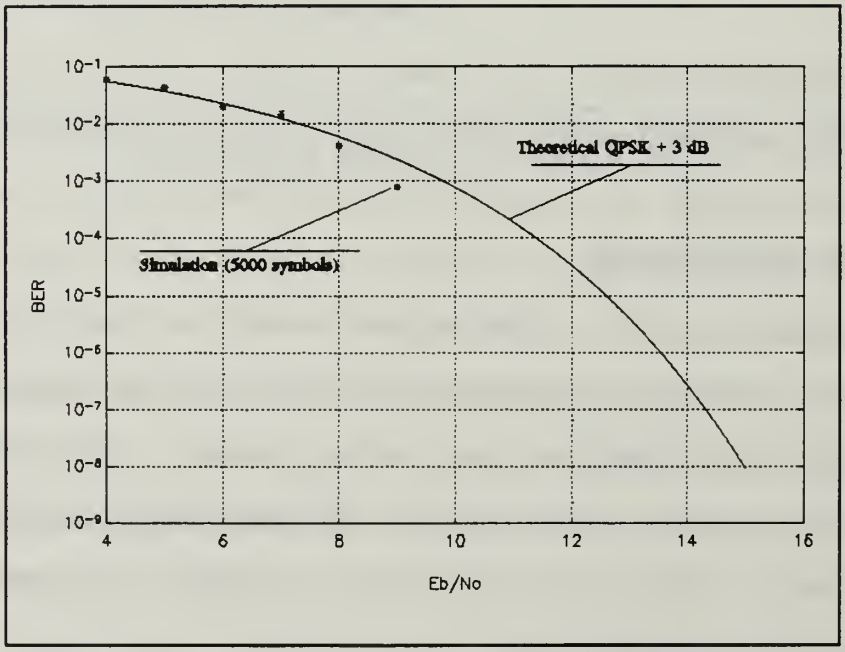

Figure 20 DQPSK Simulation in AWGN

2. Rate 1/2 Convolutionally Encoded DQPSK in AWGN Channel

This second category simulation was also a test simulation to check the convolutional encoder performance. In this case we used k=1 in (3-2) for the bit energy per information bit transferred. The signal constellation at the output of the differential demodulator for 13 dB E_b/N_o is illustrated in Fig. 21.

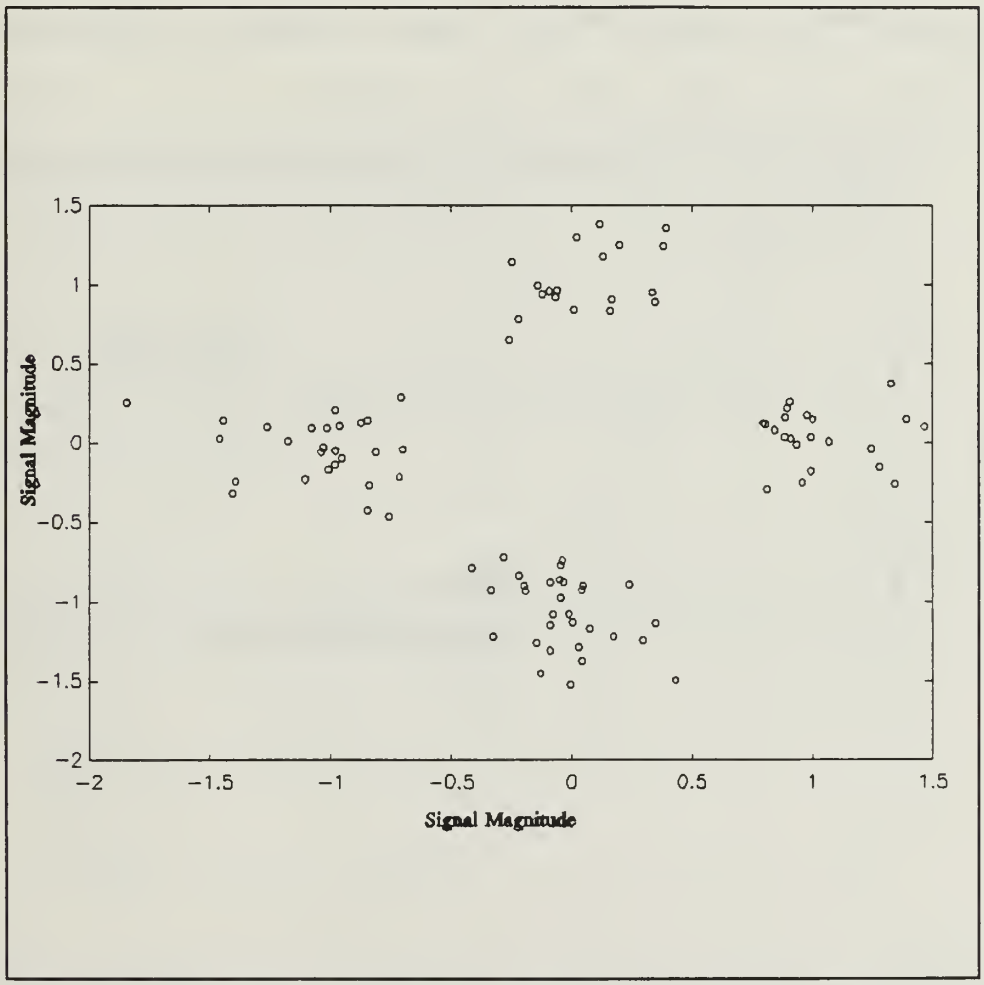

Figure 21 Signal Constellation of Convolutional Encoded DQPSK in AWGN Channel for 13 dB E_b/N_o.

The simulation results for the convolutionally encoded DQPSK in the AWGN channel are shown in Fig. 22. In the figure, BER slowly changes until it reaches the threshold of 10^{-2}, then the effect of convolutional encoding becomes dominant; we see an asymptotic performance curve as is characteristic of systems with forward error correction coding.

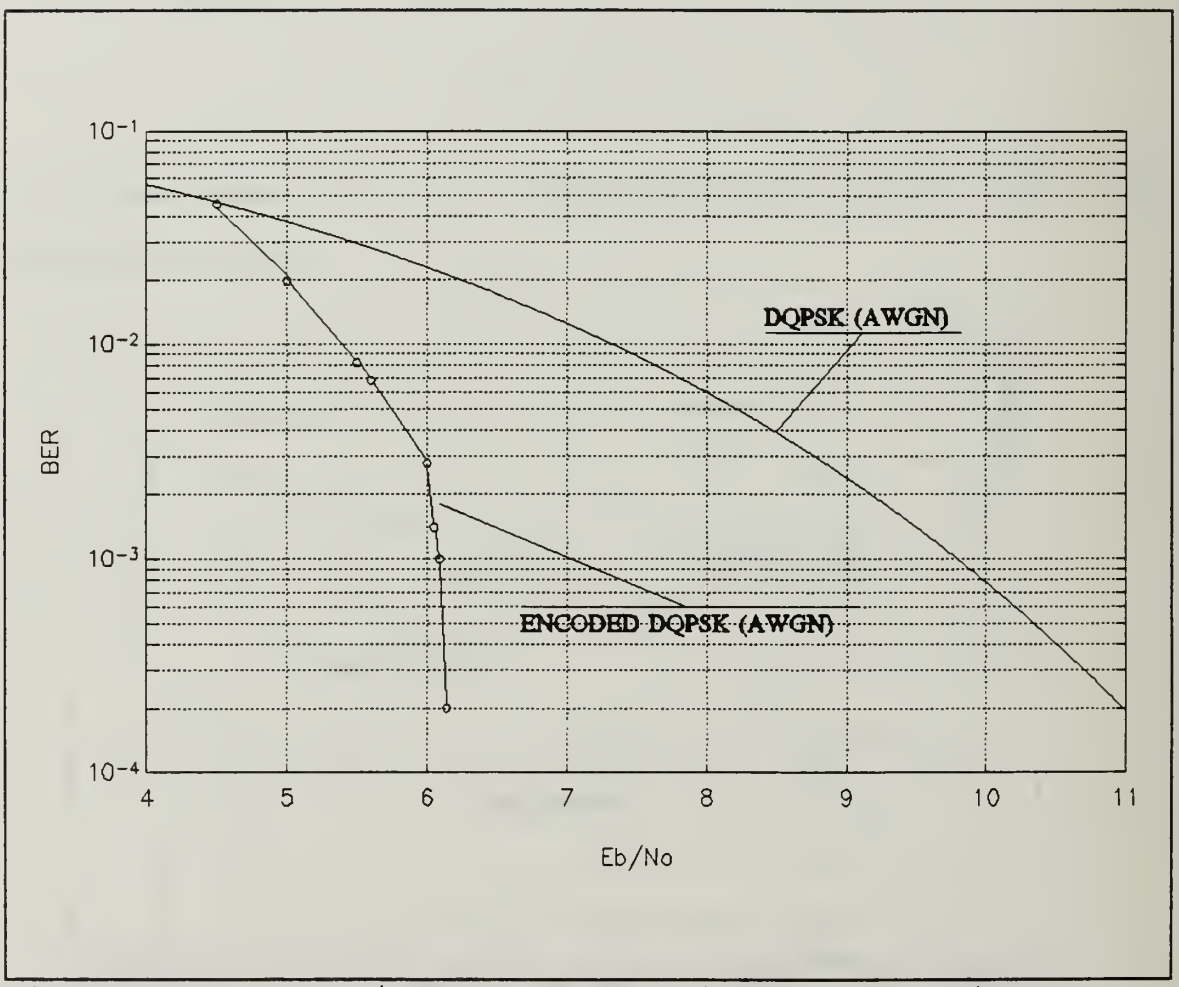

Figure 22 Rate 1/2 Encoded DQPSK in AWGN Satellite Channel

3. Rate 1/2 Convolutionally Encoded DQPSK in Fading Satellite Channel

Given the results in AWGN channel we, next cascaded our transponder components and fading generating function to the simulation program and tested the entire system for different baud rates. Figure 23 presents the signal constellation at the output of the differential demodulator for this system. Comparison of Fig. 23 and Fig. 21, constellation in AWGN, shows the effects of fading are principally on the signal amplitude which now varies in accordance with the Rayleigh probability distribution.

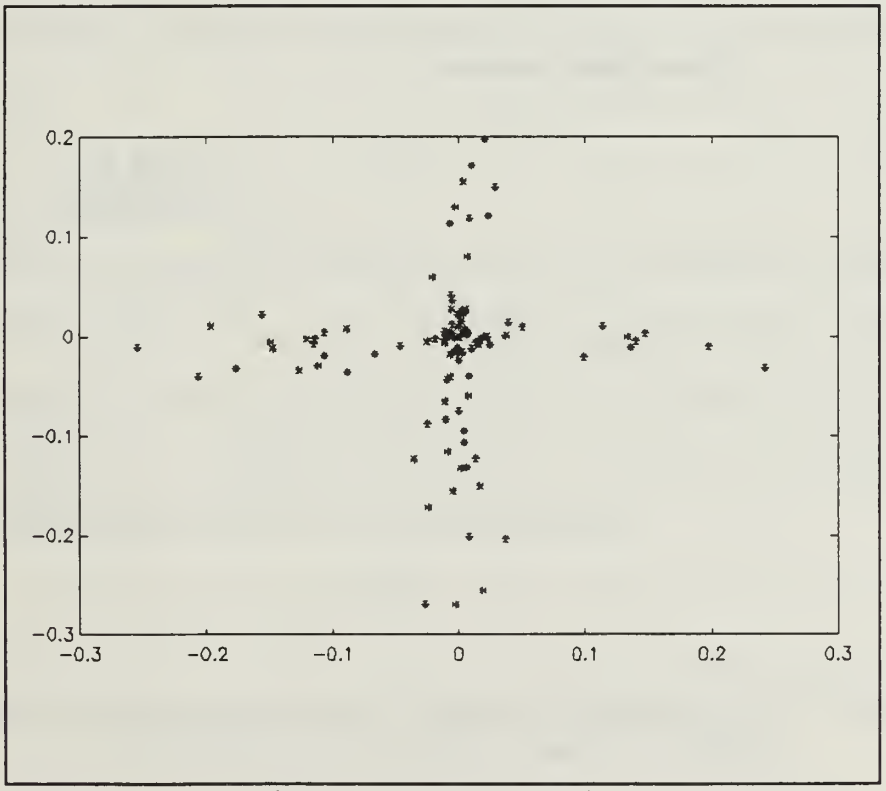

Figure 23 Signal Constellation for DQPSK in Fading Satellite Channel for 25 dB E_b/N_o.

The constellation is for E_b/N_o 25 dB. Though the effect of the noise is very small, the amplitude variation due to fading is coarse. This variation is the source of the performance degradation due to fading.

The performance of the convolutionally encoded DQPSK is illustrated in Fig. 24.

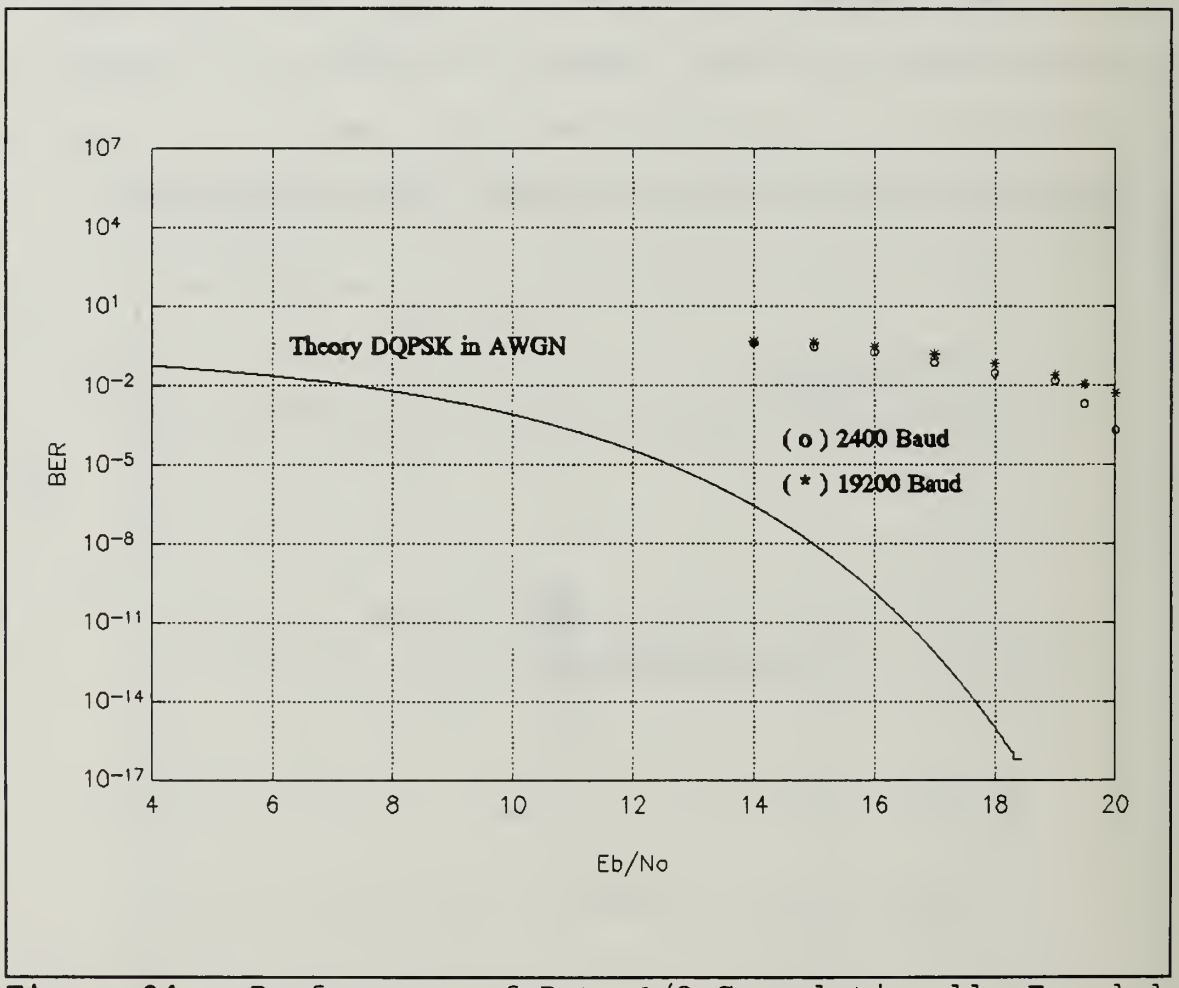

Figure 24 Performance of Rate 1/2 Convolutionally Encoded DQPSK in Fading Satellite Channel

This simulation was done with transponder components, including interleaver/deinterleaver. Interleaving depth is 0.01 of the total transferred bits. Fading was applied to make differential phase shift between symbols on the order of 10^{-2} radians. We plotted the curves related to baud rates 2400 and 19200, due to very close simulation outcomes.

If we compare these performance curves with those of the AWGN channel we see an approximate performance degradation of 13 dB. This loss is due to fading effect of the channel and is the performance that can be achieved with present fading mitigation methods. But this performance, for the same throughput, is superior to the performance of DBPSK in the fading channel which is entirely unacceptable. (See Fig. 19.)

C. MSK SIMULATIONS

We simulated MSK modulation for the satellite channel in two categories: first, for the AWGN channel with transponder components included, and then with the fading portion of the code included to observe the effect of fading on MSK.

1. MSK Performance in AWGN Channel

We used 10000 bits in these simulations. The E_b/N_o was calculated as described in (3-2). The theoretical curve of the MSK in AWGN channel is [Ref. 7: p. 549],

$$P_b = Q\left(\sqrt{2\left(\frac{Eb}{No}\right)}\right) \qquad (3-4)$$

The performance shown in Fig. 25 is the result of simulation in the AWGN channel with transponder components included.

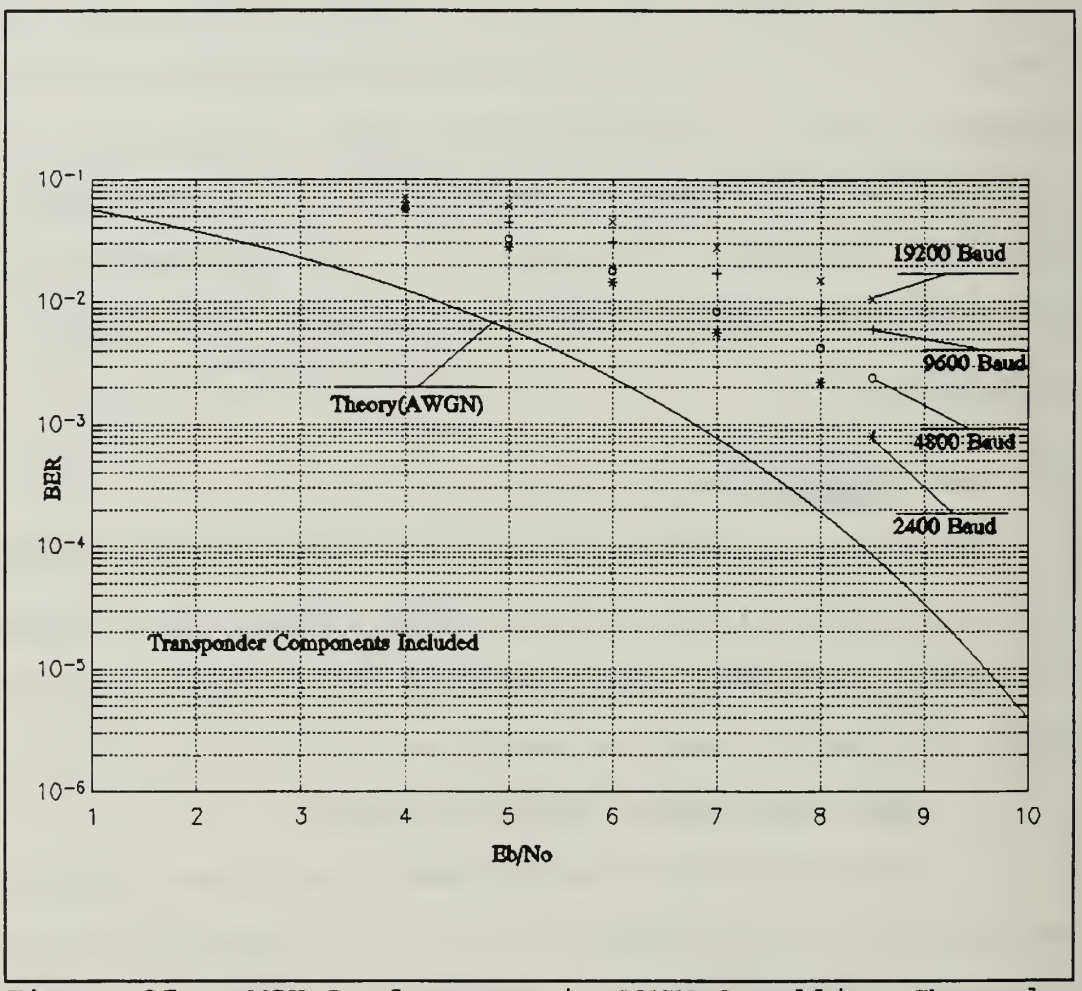

Figure 25 MSK Performance in AWGN Satellite Channel

2. MSK Simulations in Fading Satellite Channel

This second type of simulation was done in the presence of fading with transponder components included. The performance of MSK in the fading channel was severely degraded compared to that in the AWGN channel.

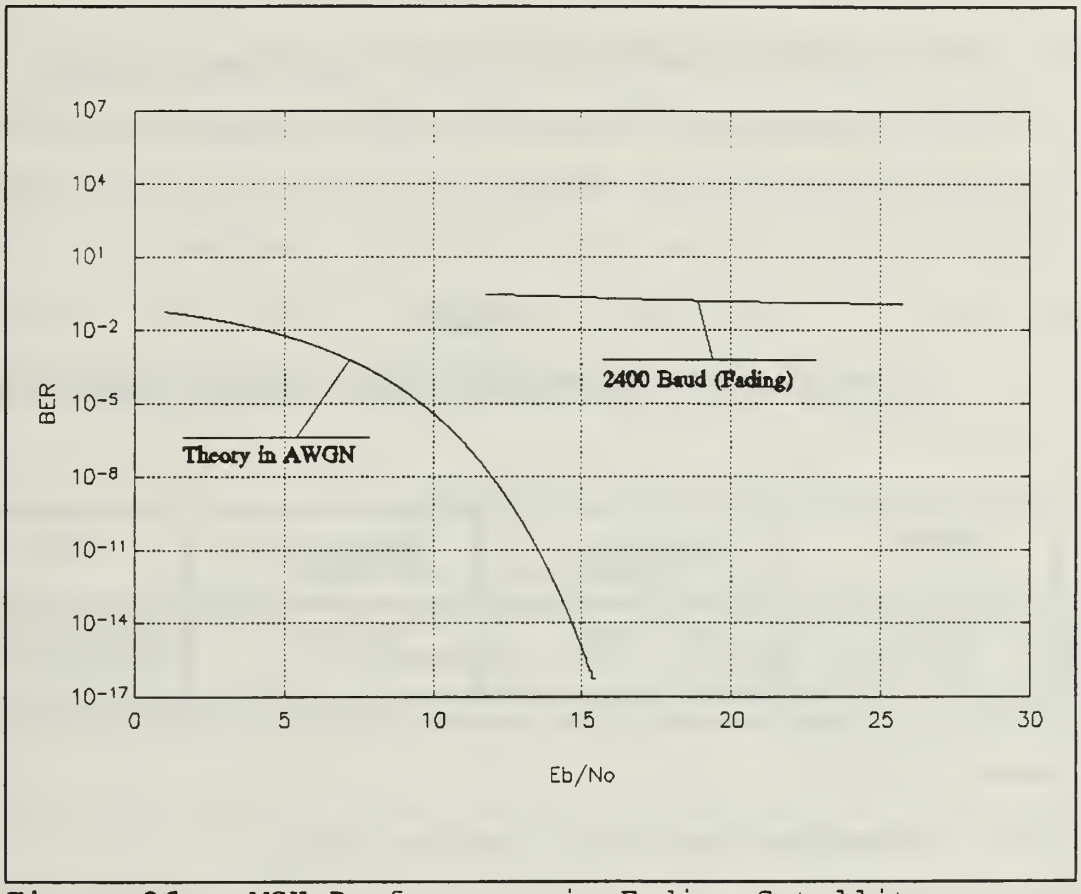

Figure 26 MSK Performance in Fading Satellite Channel

IV. LINK BUDGET MARGINS DERIVED FROM THE SIMULATIONS

In this chapter we will show the link budget margins between the calculated values of the UHF Follow-On Satellite System and the simulation results.

A. UHF FOLLOW-ON SATELLITE SYSTEM CALCULATED VALUES.

The parameters of calculating the carrier-to-noise ratios are defined in Ref. 6. These values are tabulated in Table 1.

TABLE 1

UHF FOLLOW-ON SATELLITE SPACE SEGMENT SPECIFICS

BAND	$EIRP_{RELAY}$	$EIRP_{ENHANCED}$	G/T
UHF	27 dBW	29 dBW	-14.4 dB/K

where

EIRP = Effective Isotropic Radiated Power,

G/T = Receiver Sensitivity.

For evaluation purposes we chose the AN/WSC-3, the US Navy's primary mobile UHF satellite terminal transceiver whose characteristics are defined in [Ref. 11: p. 16] and shown in Table 2.

TABLE 2

AN/WSC-3 TRANSCEIVER SPECIFICS

EIRP	G/T
22 dBW	-18.3 dB/K

We selected two extreme cases for the location of the terrestrial terminals. The first terminal was under direct projection of the satellite coverage which has an elevation angle of 90 degrees, and the other was with the minimum elevation angle of 5 degrees. For these stations we calculated their slant distances as shown in Table 3.

TABLE 3

TERMINAL SLANT RANGES

STATION	ELEVATION (E)	SLANT DISTANCE (d)
1	90°	35768 km.
2	5°	41124 km.

The corresponding free space path losses are calculated as follows;

$$L_{s_{(dB)}} = 20\log_{10}\left(\frac{4.\pi.f_c.d}{c}\right) \qquad (4\text{-}1)$$

Where

f_c : Carrier frequency (For our case 300 MHz.)

d : Slant distance (km)

c : Speed of light (2.99×10^5 km)

The free space path losses for stations 1 and 2 were calculated and found to be 173 dB and 174.29 dB, respectively.

The carrier-to-noise ratios for uplink and downlink cases were calculated by using below equations [Ref. 11: p. 74].

$$\left(\frac{C}{N}\right)_{uplink} = EIRP_{terminal} - L_s + \left(\frac{G}{T}\right)_{satellite} - 10\log_{10}k - BW \quad (4\text{-}2)$$

$$\left(\frac{C}{N}\right)_{downlink} = EIRP_{satellite} - L_s + \left(\frac{G}{T}\right)_{terminal} - 10\log_{10}k - BW \quad (4\text{-}3)$$

Where :

$EIRP_{terminal}$: EIRP of the ground terminal (dBW),

$EIRP_{satellite}$: EIRP of the satellite (dBW),

$G/T_{terminal}$: Receiver sensitivity of the ground terminal (dB/K),

$G/T_{satellite}$: Receiver sensitivity of the satellite (dB/K),

BW : Noise Bandwith ($10\log_{10}$(25 kHz)),

L_s : Free space path loss (dB),

k : Boltzmann's constant (dBW/K-Hz).

Calculated carrier-to-noise ratios for uplink and downlink are tabulated in Tables 4 and 5 respectively.

TABLE 4

UPLINK CARRIER-TO-NOISE RATIOS

STATION	$(C/N)_{UPLINK}$
1	19.2 dB.
2	17.9 dB.

TABLE 5

DOWNLINK CARRIER-TO-NOISE RATIOS

STATION	$(C/N)_{RELAY}$	$(C/N)_{ENHANCED}$
1	20.4 dB.	22.4 dB.
2	19.1 dB.	21.1 dB.

B. SIMULATED CARRIER-TO-NOISE RATIOS

The performance curves of the simulations performed in the satellite channel gave us the E_b/N_o values for corresponding probablity of error. By using (4-4) we converted our E_b/N_o values to carrier-to-noise ratios for different data rates.

$$(\frac{E_b}{N_o}) = \frac{B}{R}(\frac{C}{N}) \qquad (4-4)$$

Where :

R = Baud rate (symbols per second),

B = Noise Bandwith (25 kHz).

From this relation we calculated our simulated required carrier-to-noise ratios for BER 10^{-4} and tabulated them in Table 6.

TABLE 6

SIMULATED CARRIER-TO-NOISE RATIOS

MODULATION	CHANNEL	2400 Baud	4800 Baud	9600 Baud	19200 Baud
DBPSK	AWGN	-0.17 dB.	3.83 dB.	8.34 dB.	12.85 dB.
DBPSK	FADING	29.8 dB.	33.8 dB.	38.84 dB.	43.85 dB.
$DQPSK_{ENCOD}$	AWGN	-3.77 dB.	NO SIMULATION	NO SIMULATION	NO SIMULATION
$DQPSK_{ENCOD}$	FADING	9.82 dB.	13.3 dB.	16.84 dB.	20.44 dB.
MSK	AWGN	-0.17 dB.	3.63 dB.	7.84 dB.	11.85

For MSK in the fading satellite channel we did not calculate the corresponding carrier-to-noise ratios, because they were out of range for BER 10^{-4}.

We calculated our carrier-to-noise ratios for 19200 baud rate, but it should be noted that maximum baud rate for AN/WSC-3 is 9600 baud.

C. LINK MARGINS

If we compare the contents of Table 6 and with those of Tables 4 and 5 for the fading satellite channel, we see that the only suitable modulation technique is convolutionally encoded DQPSK, which is lower than the carrier-to-noise ratios for station 1 in all cases and for station 2 in all cases except baud rate 19200 in the relay mode of the satellite.

We tabulated link margins in Table 7 for the enhanced mode of the satellite and for the worst case of earth station, station 2, which has more slant range.

TABLE 7

LINK MARGINS FOR ENHANCED MODE

MODULATION	CHANNEL	2400 BAUD	4800 BAUD	9600 BAUD	19200 BAUD
DBPSK	AWGN	21.27 dB	17.27 dB	12.76 dB	8.25 dB
DBPSK	FADING	-8.7 dB	-12.7 dB	-17.74 dB	-22.75 dB
$DQPSK_{ENCOD}$	AWGN	24.87 dB	NO SIMULATION	NO SIMULATION	NO SIMULATION
$DQPSK_{ENCOD}$	FADING	11.28 dB	7.84 dB	4.26 dB	0.66 dB
MSK	AWGN	21.2 dB	17.47 dB	13.26 dB	9.25 dB

For the other fading situations, predicted carrier-to noise ratios are greater than the actual proposed carrier-to-noise ratios. Therefore, there is no margin for these cases.

For the AWGN channel, in all cases the simulated carrier-to-noise ratios are below the actual values of the proposal.

V. RESULTS AND RECOMMENDATIONS

For this thesis we simulated the satellite fading channel proposed in Ref. 6. We also simulated DBPSK, rate 1/2 convolutionally encoded DQPSK and MSK for both AWGN and fading satellite channels. We evaluated their performances for these channels for different baud rates and for different BER values.

The results show that the fading effect of the channel degrades the performance of the DBPSK severely. We tested a convolutional encoder and interleaver/deinterleaver pair cascaded with DQPSK modulation to mitigate the effect of fading without changing the throughput relative to DBPSK. We see that the performance of the encoder combined with DQPSK is 20 dB better than that of the DBPSK in the fading satellite channel. Additionally we simulated MSK for different baud rates in AWGN and saw that it has an approximate 1 dB loss due to baud rate increase. However, its performance in fading was severely degraded.

Our ultimate aim was to cascade this MSK modulation with a convolutional encoder and apply some diversity technique to evaluate its performance for the satellite channel due to its smooth spectral character. Follow-on studies may provide opportunities for other students to combine encoding and

continuous phase modulation (CPM) with the satellite fading channel. However, how to interleave coded CPFSK symbols is an unsolved problem.

Fading also can be modified to simulate the multi-path effect for present modulation techniques and combined coding and CPM.

Further, two channels should be simulated simultaneously to determine the effect of adjacent channel interference on the different modulation and coding schemes.

APPENDIX A. DBPSK CODES

A. DBPSK MAIN PROGRAM

```
clear;
clg
%*****************************************
%Definition of variables
%*****************************************
m=10000;            %Number of bits
del=.01             %Differential phase shift for the fading
cf=[30 15 7 4];     %Number of samples per symbol
sigma= 1.5 ;        %Standart deviation of the noise
filt_coeff_pre=.32; %Prelimiter filter coefficient
filt_coeff_post=.21;%Postlimiter filt5er coefficient

%*****************************************
%For each sample per symbol and for each noise standart
%deviation program runs once and saves the number of errors
%*****************************************

for kk=1:length(cf);
 for i=1:length(sigma);
  N=cf(kk);
  n_coef=sigma(i);
  rnd_o=msg(m);           %Random bit stream creation
  dif_o=dif_cod(rnd_o); %Differential decoding
  map_o= map(dif_o);     %Map {0,1} to {-1,1} respectively
  mod_o=modul(map_o,N); %Phase Shift Keying Modulating
  mod_o=mod_o';
  mod_o=mod_o(:);
  mod_o=exp(j*pi/4)*mod_o; %Power Splitting
  I=fading(del,(m+1)*N);   %I channel fading stream creation
  Q=fading(del,(m+1)*N);   %Q channel fading stream creation
  mod_o=I.*real(mod_o) +j*(Q.*imag(mod_o)); %Fading addition
  save m1 mod_o;
  clear rnd_o dif_o map_o mod_o I Q;
  ch_dis=awgn_ch((m+1)*N,n_coef);%AWGN  creation
  load m1;
  dem_in=ch_dis+mod_o;%AWGN addtion to the modulator output
  clear mod_o ch_dis ;
  buf=ones(2*N,1);
  dem_in=[buf;dem_in];%Filter steady state time compensation
  [b1,a1]=cheby1(6,.01,filt_coeff_pre);%Prefilter
                                       %coefficients
```

```
  dem_in=filter(b1,a1,dem_in); %Prefiltering
  dem_in=limiter(dem_in); % Hardlimiter application
  [b2,a2]=cheby1(4,.025,filt_coeff_post);%Postfilter
                                          %coefficients
  dem_in=filter(b2,a2,dem_in);%Postfiltering
  dem_in=dem_in(2*N+1:length(dem_in),:);%Shifting due to
                                        %filter
  dem_in=[dem_in(8:length(dem_in));dem_in(1:7)];%delay
  dem_in=reshape(dem_in,N,m+1);
  dem_in=dem_in';
  [out, metric]=demod(dem_in,N,m);%Demodulating
  clear dem_in;
  save o1 out;
  load r1;
  [n_er(kk,i)]=compa(rnd_o,out,m);%Comparison of received
                                  %and transmitted sequences
   end
end
save res.1 n_er %Saving number of errors
```

B. MSG.M (INPUT SEQUENCE GENERATION FUNCTION)

```
function u = msg(k)
%This M-file accepts k the number of bits that will be
%returned in the vector u for seed 1.
rand('uniform')
u = round(rand(1,k));
```

C. DIF_COD.M (DIFFERENTIAL ENCODER FUNCTION)

```
function [dif_o]=dif_cod(in)
%This M file accepts an input vector and differentially
%encodes it by inserting the reference bit 1 at the
%beginning of the vector.
a=length(in);
y=[1,zeros(1,a)];   %first 1 is the reference bit
for i=1:a;
  y(i+1)=xor(in(i),y(i));
end
dif_o=y;
```

D. MAP.M (MAPPING FUNCTION)

```
function o=map(in)
%This M file accepts the binary vector and mappes 0's to
%1's.
a=length(in);
for i=1:a
```

```
 if (in(i)==0)
  in(i)=-1;
 end
end
o=in;
```

E. MODUL.M (MODULATOR FUNCTION)

```
function [y]=modul(in,N)
%This M file accepts the differentially encoded and mapped
%vector in and N number of samples per symbol and and
%generates a matrix with dimensions (length(in) x N).
a=length(in);
y=in'*ones(1,N);
```

F. FADING.M (FADING GENERATION FUNCTION)

```
function y=fading(del,mm)
%This M file accepts differential phase shift del and total
%number of the samples mm in the input vector and creates
%mm normal distributed variables and passed these through
%two RC %filters.
s=exp(-del/2.146193);
ss=((1-s^2)^3/(1+s^2))^.25;
b=[ss];
a=[1 -s];
rand('normal');
t=rand(mm,1);
k=filter(b,a,t);
y=filter(b,a,k);
```

G. DEMOD.M (DEMODULATOR FUNCTION)

```
function [out,a]=demod(mod_o,N,m)
%This M file accepts fadign and AWGN effected modulator
%output mod_out, number of sample per symbol,and total
%number of symbols. And for each symbol takes the difference
%and sum of the present and and past symbol sums and takes
%squares of their absolute values.And if dif < sum then
%decides input bit is 0.
o=ones(1:m);
for i=2:m+1
  dif(i-1)=abs(sum(mod_o(i,:))-sum(mod_o(i-1,:))).^2;
  su(i-1)=abs(sum(mod_o(i,:))+sum(mod_o(i-1,:))).^2;
  metric(i-1)=dif(i-1)-su(i-1);
  if metric(i-1)<0;
```

```
      o(i-1)=0;
  end
end
a=metric;
out=o;
```

APPENDIX B. DQPSK CODES

A. DQPSK MAIN PROGRAM

```
clear
clg
del=0.1;
m=10000;   %Number of symbols that wil be transmitted
%*********************************************
%You can change number of samples N accoeding to the
%Baud rate you desire to work on.N=30 for 2400,N=15
%for 4800,N=7 for 9600 and N=4 for 19200 Baud.You have
%to consider to change the standart deviation of AWGN for
%each case.
**********************************************
N=30;%Samples per symbol (This is for 2400 baud)
load fec    %Data file that has G64 and T64
u=msg(0,m); %Random symbol creation
j=sqrt(-1);
save utemp1 u
v=cncd(2,1,6,G64,u,zeros(1,12));%Convolutional encoding
mm=bm(2,v); %Binary to M-ary conversion
clear v u;
mm=inter(50,200,mm);%Interleaving (50 by 200)
M=dphmod(2,mm);%Differential encoding
clear mm ;
M=resamp(N,M);%Generate each element of M , N times
M=reshape(M,N,m+1);%Reshape to build (m=1) by N matrix
M=conj(M');
save Mtemp M;
clear M;
sigma_v=[1.2247 ];%AWGN standart deviation
for ss=1:length(sigma_v)
     Np=80;  % This is the path history length for the
             %viterbi  decoder
     KK=fad(del,m+1,N);%Applying fading
     load Mtemp;
     M=KK.*M;
     clear  KK;
     R=awgn(M,sigma_v(ss));%AWGN addition
     R=conj(R');
     R=R(:);%Make it vector to pass it from filter
     clear M;
     buf=[ones(2*N,1)];
     R=[buf;R];
     [b1,a1]=cheby1(6,.01,.32);
     R=filter(b1,a1,R);
```

```
      R=limiter(R);
      [b2,a2]=cheby1(4,.025,.21);
       R=filter(b2,a2,R);
       R=R(2*N+1:length(R));
       R=[R(8:length(R));R(1:7)];
       R=reshape(R,N,m+1);
       R=conj(R');
       for i=1:m+1
           temp(i,:)=xcorr(ones(1,N)*exp(j*pi/4),R(i,:));
           %This correlates each symbol with ones to find
           %metrics
           res(i)=(temp(i,N))/N;%Normalization
       end
       clear temp R ;
       [s,R]=ddmod(2,2,res);%Differential demodulator
       clear res s;
       R=dinter(50,200,R); %Deinterleaving
       D=eucdis(2,R);%Euclidian distance finder
       clear R temp2;
       [rr,cc]=size(D);
       PH=zeros(64,3*Np);
       for h=1:rr
           PH=viterbi(1,Np,PH,T64,D(h,:));%Viterbi decoding
           ud(h)=PH(1,240)-1;%Signal level offset
       end
       clear PH D;load utemp;
       [a,b]=mtrxv(79,u,ud);%Shifting due to path history
                            %length.
       clear u ud;
       err(ss)=check(a,b);%Error check
       clear a b;
end
save result err
```

B. CNCD.M (CONVOLUTIONAL ENCODER FUNCTION)

```
function [v,vr] = cncd(n,k,m,Gr,u,r)
%                  CONVOLUTIONAL ENCODER
%                    Paul H. Moose
%                 Naval Postgraduate School
%                     04-24-91
%
%   This m-file is a feedforward convolutional encoder for
%an(n,k,m) linear  convolutional code.It codes binary vector
%u of length Lk bits into binary vector v of length Ln bits
%plus a residual vector vr of nm bits.The code is determined
%by the matrix filter Gr of dimension k rows by n(m+1)
%columns.(See page 292 in  Lin and Costello Gr=[G0 G1 G2
%Gm].).
%   A previous initial vector r of nm bits is added to the
%first nm bits of v.
```

```
%   The program first checks to see if Gr has correct
%dimensions.If it does, it continues. If not, it ends and
%returns an empty v.
[kg N]=size(Gr);
if N == n*(m+1)
%Matrix has correct no of columns
if kg == k
%Matrix has correct no. of rows
%The program now checks to see if u has a multiple of k
%bits.If so, it continues, if not it adds zeros to u to make
%it so.
[zz Lk]=size(u);
q=rem(Lk,k);
    if q==0
    u=u;
    else
    u= [u zeros(1,k-q)];
    [zz Lk]=size(u);
    end
% We now have a msgword u with Lk bits, an exact multiple of
%k bits. The output code word is formed by finding first the
%output for each  word of k bits. These outputs are make the
%rows of w.
L=round(Lk/k);
    for j=1:L
    w(j,:)=rem(u(:,(j-1)*k+1:j*k)*Gr,2);
    end
%The output vector is formed by adding rows of w with
%progressive delay of n bits for each row. The initial
%condition vector r is placed in the first nm bits of v.
v=[r zeros(1,n*L)];
        for l=1:L
v(:,n*(l-1)+1:(m+l)*n)= v(:,n*(l-1)+1:(m+l)*n) +w(l,:);
        end
v=rem(v,2);
%This vector is divided into the residal portion of nm bits
%and the first nL bits.
vr=v(:,n*L+1:n*(L+m));
v=v(:,1:n*L);
%This is the end of the program for v.
else,break,end  %End if wrong no. of rows
else,break,end  %End if wrong no. of columns
```

C. BM.M (BINARY TO M-ARY FUNCTION)

```
function m = bm(q,v)
%                    BINARY TO M-ARY CONVERTER
%                       Paul H. Moose
%                   Naval Postgraduate School
%                         06-10-91
```

```
% The bits are stripped q at a time and are mapped
% to a symbol vector m with integer values 0 to 2^q-1.
% Zeros are added at the end of v ,to have integral q
% if necessary.
N=2^q;
[m n]=size(v);
r = rem(n,q);
if r == 0
v = v;
else
v = [v zeros(1,q-r)];
end
% We now have a codeword v with an exact multiple of q bits.
% The bits must now be stripped q at a time and used to
%generate the m-ary values.
map = 1;
for j =1:q-1
map =[map 2^j]; % This makes the least significant bit on
                % the left
end
[m n] = size(v);
p =round(n/q);
for i= 1:p
s = v(:,(i-1)*q+1:(i-1)*q+q);
m(i) =s*map';
end
```

D. INTER.M (INTERLEAVER FUNCTION)

```
function mi=intlv(l,k,m)
%This m-file is a block interleaver. The matrix m is read
%into a column by rows, then the column is read into
% a (l,k) matrix by columns and read out into vector mi by
%rows. mi is shaped into matrix of same dimension as m.
m=conj(m');
y=m(:);
N=length(y);
if l*k==N
x=zeros(l,k);
x(:)=y;
x=conj(x');
yi=x(:);
m(:)=yi;
mi=conj(m');
end
```

E.DPHMOD.M (DIFFERENTIAL PHASE MODULATOR)

```
function MD = dphmod(q,m)
%This M-file creates complex values MD with amplitude one
%and one of 2^q equal phase values. The first value is one.
%The next values are differentially coded in phase. The
%input symbols  are in the m vector.
N=2^q;
dph=2*pi/N;
[rr n]=size(m);
for k=1:rr
md=cumsum(m(k,:)); %This differentially codes the phase
                   %values
i =sqrt(-1);
MD(k,:) = exp(i*dph.*md);
end
MD=[ones(rr,1) MD];
```

F. RESAMP.M (SAMPLING FUNCTION)

```
function y =resamp(k,x)
%This M-file resamples the vector x to generate y. If k is
%negative,every kth sample is put into the vector y. If k is
%positive, every sample in x is repeated k times in y.
[m n]=size(x);
if k >0
for i=1:n
    for j=1:k
    y(i*k-j+1)=x(i);
    end
end
else
k =abs(k);
l = fix(n/k);
for i = 1:l
y(i)=x((i-1)*k+1);
end
end
```

G. FAD.M (FADING FUNCTION)

```
function y=fading(del,aa,bb);
%This M file accepts del, the differential phase shift and
%the dimensions of the modulator output and calculates
%filter coefficients for 2 RC filters, and generates 2
%normal random distributed sequences passes these sequences
%through the filters and forms the Rayleigh fading signal.
s=exp(-del/2.146193);
```

```
ss=((1-s^2)^3/(1+s^2))^.25;
b=[ss];
a=[1 -s];
rand('normal');
rand('seed',0);
k=rand(aa*bb,1);
k=filter(b,a,k,1);
k=filter(b,a,k,1);
z=rand(aa*bb,1);
z=filter(b,a,z,1);
z=filter(b,a,z,1);
k=reshape(k,aa,bb);
z=reshape(z,aa,bb);
j=sqrt(-1);
y=k+j*z;
```

H. AWGN.m (AWGN FUNCTION)

```
function y = awgn(X,sigma)
% Awgn is an m-file that adds awgn noise to the matrix x
% Eb/No=1/(2*sigma^2).
[rr,cc]=size(x);
rand('normal')
w=rand(rr,cc)+i*rand(rr,cc);
y=x+sigma.*w;
```

I. DDMOD.M (DIFFERENTIAL DEMODULATOR FUNCTION)

```
function [s,M] =ddmod(qp,q,MD)
%This M-file differentially demodulates complex modulation
%values in MD  into 2^qp equal phase sectors from
%constellations of 2^q phase sectors
%sectors into ka amplitude bits. the output is [s M] where
% s is the phase sector number and M is the differentially
%decoded  modulation values.
k=length(MD);
for j=1:k-1
M(j)=MD(j+1)*conj(MD(j));
end
N=2^qp;
dph=2*pi/N;
phn=angle(M)./dph;
s=rem(round(phn)+N,N);
```

J. DINTLV.M (DEINTERLEAVER FUNCTION)

```
function m=dintlv(l,k,mi)
%This m-file is a block deinterleaver. The vector mi is
%read into a (k,l) matrix by columns and read out into
%vector m by rows.
mi=conj(mi');
y=mi(:);
N=length(y);
if l*k==N
x=zeros(k,l);
x(:)=y;
x=conj(x');
yi=x(:);
mi(:)=yi;
m=conj(mi');
end
```

K. EUCDIS.M (EUCLIDIAN DISTANCE FUNCTION)

```
function D = eucdis(q,R)
%                       EUCLIDEAN DISTANCE METRICS
%                            Paul H. Moose
%                       Naval Postgraduate School
%                              06-17-92
%
% This M-file finds Euclidean distance of elements
% in vector R from 2^q unit amplitude vectors
% equally spaced on the unit circle. It stores these as rows
%of D.
N=2^q;
L=length(R);
index=1:N;
dph=2*pi/N;
M0=exp(j*dph.*(index-1));
for l=1:L
D(l,:)=abs(R(l).*ones(M0)-M0);
end
```

L. VITERBI.M (SOFT VITERBI DECODING FUNCTION)

```
function PHN = viterbi(k,Np,PH,T,D)
%               SOFT VITERBI DECODER
%          Paul H. Moose
%           Naval Postgraduate School
%                   10-09-91
%
%   This m-file decodes k bit msgwords from  2^n real
%metrics in the row vector D. (These  may, for example,
```

```
%represent the "Euclidean distance" of the received
%modulation value from each of 2^n modulation values.)
%The state transition information for a K state trellis is
%stored in a K by 3*2^k  matrix T. Each of the 2^k paths
%entering each state has its source state (one of K), path
%msgword (one of 2^k) and path codeword or signal number
%(one of 2^n) listed in the state row of T.
%   The Np length path histories are kept in matrix PH which
%is K by 3*Np. The path history for each state contains
%source state, path  weight and path msg word for Np
%previous states.
%   The output, PHN the new path history, is the update of
%PH.The decoded msg word is in the last column of PHN. (They
%should all have "merged " to agree on a common msg).
% The past histories, which are in PH, are undated on the
%basis of the "minimum weight". You can change this to the
%"maximum weight" if desired as indicated in the comments in
%the code if approriate for the metric of your receiver.
% Prior to using the function for the first symbol, you must
%initialize it with a PH matrix of dimension K by 3*Np. (I
%usually use an all zero matrix). For the second and
%following symbols, PH is made the PHN from the previous
% execution of the viterbi function.
P=PH(:,2)';
wt=T(:,3:3:3*2^k)'; ux=T(:,1:3:3*2^k-2)';
aa=D(wt(:)); bb=P(ux(:));
wt(:)=aa+bb; %This contains all weights(columns) for each
             %state(row)
[ a,b]=min(wt); % Use max(wt) here for maximum
X(:,2)=a';
X(:,1)=diag(T(:,3.*b-2));
X(:,3)=diag(T(:,3.*b-1)); %Stores path msgword. Chg 3.*b-1
%to 3.*b to keep path codewords instead of msg words.
% Now  append old paths to new paths to get survivors.
PHN=[X PH(X(:,1),1:3*Np-3)];
```

M. MTRXV.M (OFFSET FUNCTION)

```
function [a,b]=mtrxv(N,A,B)
%This changes matrices A and B to vectors a and b with
%offset N.
A=A'; B=B';
a=A(:); b=B(:);
a=a'; b=b';
if length(a)==length(b)
a=a(1:length(a)-N);b=b(N+1:length(b));
end
```

N. CHECK.M (ERROR CHECKER FUNCTION)

```
function [Nb,I,e] =check(x,y)
%                      SYMBOL ERROR CHECK
%                        Paul H. Moose
%                    Naval Postgraduate School
%                         09-01-91
%
% This m-file locates the positions in vectors x and y that
%do not  agree. It returns a one in e if they do not agree
%and a zero if they do agree. e is the error vector if x and
%y are binary. I is a vector of error location numbers. Nb
%is the sum of the elements of e.
e=(x~=y);
I=find(e);
Nb=sum(e);
```

APPENDIX C. MSK CODES

A.MSK.M (MSK MAIN PROGRAM)

```
clear
clg
f=100;
T=0.015; %Symbol Period
t=0.0005;%Sample period
h=1/2;   %Modulation index
m=5000;  %Number of samples
msg=random(1,m);%Transmitted binary stream
save msg1 msg;
msg=mapper(msg);
mod_output=modulato(msg,T,t,h);
save modout1 mod_output
clear mod_output
viterbi_path_matrix=[1 0 1 2 1 4;1 1 2 2 0 3];
coeff=[2.4841 2.2139 ]; %AWGN standart deviations
for cnt=1:length(coeff);
  load modout1;
  [aa,bb]=size(mod_output);
  [awgn_noise]=awgn_ch(aa,coeff(cnt),bb);
  noise=awgn_ch(aa,coeff(cnt),bb);
  mod_output=mod_output+noise;
  clear noise;
  mod_output=conj(mod_output');
  mod_output=mod_output(:);
  buf=ones(2*bb,1);
  mod_output=[buf;mod_output];
  clear buf;
  [b1,a1]=cheby1(6,.01,.32);
  mod_output=filter(b1,a1,mod_output);
  mod_output=limiter(mod_output);
  [b2,a2]=cheby1(4,.025,.21);
  mod_output=filter(b2,a2,mod_output);
  mod_output=mod_output(2*bb+1:length(mod_output),:);
   mod_output=[mod_output(8:length(mod_output));mod_output(1:7)];
  mod_output=reshape(mod_output,bb,aa);
  mod_output=conj(mod_output');
  [demodulator_output]=demodula(f,T,h,mod_output,t);
  clear mod_output;
  TT=zeros(2,60);
  for x=1:m
     D=demodulator_output(x,:);
     [TT]=softv(1,1,20,TT,viterbi_path_matrix,D);
     received_signal(x)=TT(1,60);
```

```
  end
  clear demodulator_output;
  load msg1;
  [received_signal,msg]=mtr(19,msg,received_signal);
  k=received_signal~=msg;
  nofer(cnt)=sum(k);
end
save result nofer
```

B. RANDOM.M (SYMBOL SEQUENCE GENERATOR)

```
function [random_matrix]=random(m,n);
%This M file generates m by n matrix of random binary
%values {1,0}
rand('uniform');
rand('seed',0);
a=rand(m,n);
b=ones(m,n)*0.5;
random_matrix=floor(a+b);
```

C.MAPPER.M (MAPPING FUNCTION)

```
function [mapper_output]=mapper(in);
%Transfer 0 values to -1
k=in==0;
k=-k;
mapper_output=k+in;
```

D.MODULATO.M (MODULATOR FUNCTION)

```
function y=modulato(in,T,t,h)
%This is the implementation of the CPFSK for h=1/2.
teta0=0;
cnt=1;
for s=1:length(in) %For the number of samples
    I=in(cnt);
    if cnt==1
      teta=0;%Initial theta
    else
      teta0=teta0+in(cnt-1);%Accumulation of theta0
      teta=pi*h*teta0;
    end
    time=0:t:T;
    time=time+(cnt-1)*T;
    th=pi*I*h*(time/T-(cnt-1))+teta;
    mod_output(s,:)=(exp(i*th));
    cnt=cnt+1;
end
y=mod_output;
```

E. DEMODULA.M (DEMODULATOR FUNCTION)

```
function [demod_output]=demodulator(f,T,h,mod_output,t);
%This M file accepts modlator output and matches it with
%recreated phase samples as an input the euclidian distance
%to the euclidian distance finder and returns metric for the
%soft viterbi decoder.
match_out=[];
map=[-1 1];
[m,a]=size(mod_output);
time=0:t:T;
for sample=1:m

mod_output(sample,:)=mod_output(sample,:)*exp(j*(sample-1)*p
i/2);
  for m_ary=1:2
      xx=exp(i*time*pi*h*map(m_ary)/T);
      match1=xx*conj(mod_output(sample,:)');
      match_out=[match_out match1];
  end
demod_output(sample,:)=real(match_out);
match_out=[ ];
end
R=(j*demod_output(:,1)+demod_output(:,2));
[demod_output]=eucdis(2,R);
```

F. SOFTV.M (SOFT VITERBI DECODER)

```
function PHN = softv(k,K,Np,PH,T,D)
%               Soft Viterbi Decoder
%                  Paul H. Moose
%            Univ. degli Studi di Padova
%                    17-05-91
%
%   This M-file decodes k bit msgwords from  2^n real
%metrics (These  may, for example, represent the  "distance"
%of the received modulation value from each of 2^n
%modulation values.) The state transition information for a
%2^K state trellis is in the 2^K by 3*2^k  matrix T. Each of
%the 2^k entering paths to each state has its source state
%(one of 2^K), path msgword (one of 2^k) and path codeword
%(one of 2^n) listed in the state row.
%   The path histories are kept in matrix PH that is 2^K by
%3*Np. % The path history for each state contains source
%state, path  weight and path codeword  for Np previous
states.
%   The output PHN is the update of PH, the new path
%history. The decoded code word is in the last column of
```

```
%PHN.(They should  "merge".
%   The past histories are undated on the basis of the
%"minimum  metric". You can change this to the "maximum
%metric" if desired as  indicated in the comments in the
%code.
for j=1:2^K
    X(j,2)=D(T(j,3))+PH(T(j,1),2); %path weight
    X(j,1)=T(j,1);%path source state
    X(j,3)=T(j,2);%path code word T(j,3).Chg to T(j,2) for
                  %msgword.
        for l=2:2^k
          wt = D(T(j,3*l)) +PH(T(j,3*l-2),2);
          if wt < X(j,2) %The < selects min metric
             X(j,2)=wt;
             X(j,1)=T(j,3*l-2);
             X(j,3)=T(j,3*l-1); %Chg to T(j,3*l) for
                                %codeword.
          else
          end
        end
% We now need to append old paths to new paths to get
%survivors.
PHN(j,:)= [X(j,:) PH(X(j,1),1:3*Np-3)];
end
```

LIST OF REFERENCES

1. Tri T. Ha, *Digital Satellite Communications*, McGraw Hill, New York, NY, 1990.

2. Robert L. Bogush, *Digital Communications in Fading Channels: Modulation and Coding*, Mission Research Corporation, 1990.

3. John G. Proakis, *Digital Communications* ,McGraw Hill,New York, NY, 1989.

4. Ezio Biglieri, Dariush Divsalar, Peter J. McLane, Marvin K. Simon, *Introduction To Trellis-Coded Modulation With Applications*, Macmillan, New York, NY, 1991.

5. Gottfried Ungerboeck, "Channel coding with multilevel/ phase signals," *IEEE Trans. Information Theory*, vol. IT-28, pp. 55-67, Jan. 1982.

6. Hughes Aircraft Company Space and Communications Group,"Proposal for UHF Folow-on Communications Satellite, Volume II, " Jan. 1988.

7. Leon W. Couch II, *Digital and Analog Communication Systems*, Macmillan, New York, NY, 1990.

8. Richard E. Blahut, *Digital Transmission of Information*, Addison-Wesley, Menlo Park, CA, 1990.

9. M. G. Mulligan, "Rate 2/3 convolutional coding of CPFSK," Master's thesis, University of Virginia, Charlottesville, VA, Dec. 1982.

10. A.J. Viterbi, "Error bounds for convolutional codes and asymptotically optimum decoding algorithm," *IEEE Trans. Information Theory,* vol IT-13, pp. 260-269, Apr. 1967.

11. Aaron J. Godeaux, "An evaluation of models simulating the effects of a nuclear explosion on the communication links of the UHF follow-on satellite," Master's thesis, Naval Postgraduate School, Monterey, CA, Sep. 1992.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library Code 52 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
4.	Professor P. H. Moose, Code EC/Me Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
5.	Professor T. T. Ha, Code EC/Ha Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5000	1
6.	Deniz Kuvvetleri Komutanligi Personel Daire Baskanligi Bakanliklar, Ankara, Turkey	1
7.	Deniz Harp Okulu Komutanligi Tuzla, Istanbul, Turkey	2
8.	Gölcük Tersanesi Komutanligi Gölcük, Kocaeli, Turkey	2
9.	Taskizak Tersanesi Komutanligi Taskizak, Istanbul, Turkey	2


DEMCO